



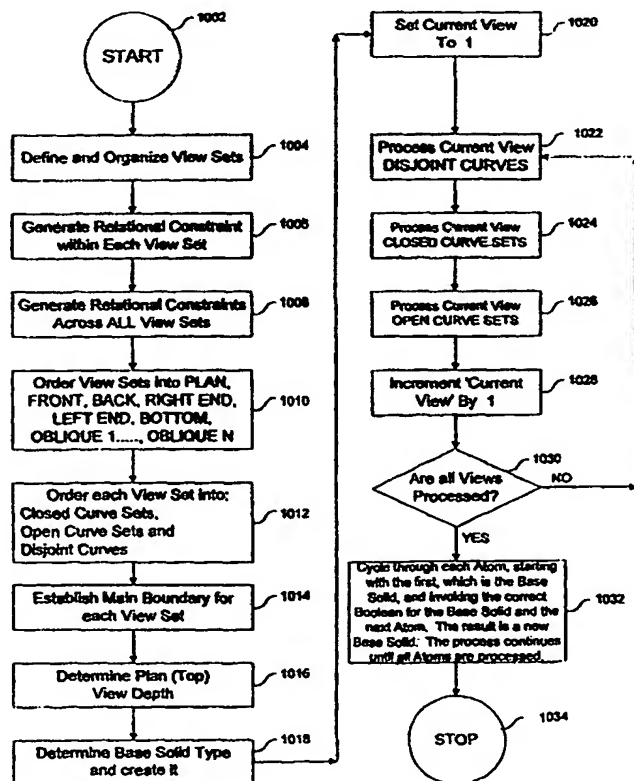
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G06T 17/00, 17/40, G06K 9/36		A1	(11) International Publication Number: WO 99/13433
			(43) International Publication Date: 18 March 1999 (18.03.99)
(21) International Application Number: PCT/US98/19189		(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).	
(22) International Filing Date: 11 September 1998 (11.09.98)			
(30) Priority Data: 08/928,831 12 September 1997 (12.09.97) US			
(71)(72) Applicant and Inventor: HANRATTY, Patrick, J. [US/US]; 2560 Liveoak Drive, Prescott, AZ 86301 (US).			
(74) Agent: POTE, Daniel, R.; Snell & Wilmer L.L.P., One Arizona Center, 400 E. Van Buren, Phoenix, AZ 85004-0001 (US).		Published With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.	

(54) Title: AUTOMATICALLY GENERATING THREE-DIMENSIONAL SOLIDS FROM TWO-DIMENSIONAL VIEW SETS

(57) Abstract

Techniques are developed for defining an engine which has as its input two-dimensional drawing views (1010), wherein the engine produces a three-dimensional geometric solid (1032). The engine comprises, among other things, a set of rules for identifying particular elements (1004) within the various two-dimensional views, and for determining the useful relationships between elements, e.g., arcs, lines and the like, in one view vis-a-vis related elements in another view (1008), to thereby ascertain the three-dimensional form defined by these elements and their corresponding relations to other elements (1032). A three-dimensional solid is thus rendered.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav	TM	Turkmenistan
BF	Burkina Faso	GR	Greece		Republic of Macedonia	TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's	NZ	New Zealand		
CM	Cameroon		Republic of Korea	PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakhstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LJ	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

"Automatically Generating Three-Dimensional Solids From Two-Dimensional View Sets"

5 **Technical Field**

The present invention relates, generally, to computer aided design/drafting (CADD) and computer aided manufacturing (CAM) systems and, more particularly, to a system for analyzing the information associated with a plurality of two-dimensional views of an object to thereby generate a true, three-dimensional geometric solid.

10 **Background Art and Technical Problems**

The pervasiveness of CADD/CAM systems over the last thirty years has revolutionized the design and manufacturing industries worldwide. Specifically, the wide availability of advanced modeling tools since the early 1970's has enabled engineers and designers to design, modify, and fabricate three-dimensional objects quickly and economically. Toward this end, much emphasis has been placed on the manufacturing end of the process -- i.e., the task of using three-dimensional design data to create prototypes, molds, and finished products via numeric control techniques (NC) and stereo lithography. The actual design and creation of three dimensional models, however, is in many respects problematic, and can best be understood in the context of the basic principles of three-dimensional design.

Solid modeling packages traditionally use one of two solid generation modes: constructive solid geometry (CSG), and boundary representation (B-REP). The CSG approach involves using the three main operations defined by George Boole (thus known as Boolean operations), namely: "union", "difference", and "intersection". These Boolean operations are applied to pairs of objects at a time. Thus, as is well known in the art, the union operation is used to add the volumes of two solids, the difference operation is used to subtract one volume from another, and the intersection operation is used to define a volume consisting only of the region in space occupied by both solids. For example, with reference to Figure 1A, a cylindrical rod 104 and block 102 are shown along with the

various solids that result from each of the three Boolean operations -- i.e. difference (solid 106), union (solid 108), and intersection (solid 110).

In contrast, The B-REP approach represents a solid as a set of vertices, edges, and faces, wherein the faces completely enclose a volume. For example, referring now to Figure 1B, a three-quarter cylinder 112 is represented by four surface elements: three planar surfaces (top face 114, bottom face 116, and front plane face 118) and one cylindrical surface 120. The topology of this solid can be represented using a linear graph 122 wherein the nodes of the graph represent the faces and the connections represent shared edges.

There are four primary methods of generating basic solids: axial sweeps, rotational sweeps, blended solids, and free-form solids. Axial sweeps are solids defined by the volume created by translation of a planar object axially through space. For example, a cylinder can be created through axial sweep of a circle. Rotational sweeps are solids defined by the volume created by rotation of a planar object through space around a line. Blended solids are solids defined by a series of closed curve sets, wherein either straight or smooth blends are used to transition between each successive curve. Free form solids are usually generated by fitting a dense set of points, or by meshing two orthogonal curve sets, to create a skin which encloses the volume. A fifth category, analytic solids, consists of boxes (right rectangular hexahedron), right cylindrical rods, cones, spheroids, and toroids. These solids are all special cases of the axial and rotational sweep.

Further information regarding these and other solid generation methods as used in a CADD/CAM environment can be found in a number of standard texts, for example: HANDBOOK OF SOLID MODELING (LaCourse ed., 1995) and Martti Mäntylä, AN INTRODUCTION TO SOLID MODELING (1988). The contents of these texts are hereby incorporated by reference.

In addition to the references listed above, well-known subroutines and equations related to basic solid geometry can be found in a number of standard texts, for example: Korn & Korn, MATHEMATICAL HANDBOOK FOR SCIENTISTS AND ENGINEERS (1961); Gerald Farin, CURVES AND SURFACES FOR COMPUTER AIDED GEOMETRIC DESIGN (2d ed., 1990); Su Bu-Qing & Liu Ding-Yuan, COMPUTATIONAL GEOMETRY: CURVE AND SURFACE MODELING (1989); Peter Lancaster & Kestutis Salkauskas, CURVE AND SURFACE FITTING (1986); Tom Lyche and Larry L. Schumaker, MATHEMATICAL METHODS IN

COMPUTER AIDED GEOMETRIC DESIGN (1989). These texts are hereby incorporated by reference.

Despite the extensive use of software-based systems for facilitating the design of articles of manufacture, existing software-based design tools generally focus on the development of a set of two-dimensional views, leaving it to the experience, intuition, and expertise of the machinist to create a three-dimensional object from the information set forth in a plurality of two-dimensional representations of the object.

Other presently known systems permit a designer to prepare a three-dimensional drawing in simulated three-dimensional space; however, the systems are typically cumbersome, and are neither intuitive, easy to learn, nor particularly user-friendly once learned. Moreover, such three-dimensional design packages do not adequately take into account the fact that most qualified designers are trained in a two-dimensional paradigm; that is, the engineering and design curricula extant today are built around various teaching and training models which, although mature and powerful, are unambiguously two-dimensional in focus and orientation.

At least one study has shown that approximately 70% of all three-dimensional CADD system users employ these systems in a two-dimensional mode -- that is, they perform three-dimensional design by creating multiple two-dimensional drawings which are later converted by human machinists into a three-dimensional object. As a result, designers and engineers have accumulated vast inventories of two-dimensional drawings and data which cannot be conveniently converted into three-dimensional objects.

Methods have been developed to facilitate, on a limited scale, generation of three-dimensional objects using two-dimensional data as input; see, for example: *Suzuki*, U.S. Patent No. 5,428,715, issued Jun. 27, 1995; *Azarbayejani et al.*, U.S. Patent No. 5,511,153, issued Apr. 23, 1996; *Kurashige et al.*, U.S. Patent No. 5,363,476, issued Nov. 8, 1994; *Abrams et al.*, U.S. Patent No. 5,587,913, issued Dec. 24, 1996; and *Niu et al.*, U.S. Patent No. 5,561,748, issued Oct. 1 1996. However, these methods are concerned generally with projection, extrusion, or mapping of two-dimensional data within three-space rather than the substantially more vexing problem of constructing solid objects from an arbitrary set of two-dimensional views.

Systems and methods are therefore needed which overcome the shortcomings of the prior art and which facilitate the conversion of two-dimensional drawings into three-dimensional object data.

Summary of the Invention

5 The present invention provides systems and methods for parsing, interpreting, and manipulating two-dimensional views of a three-dimensional object and, through the use of techniques which mimic the machinist's experience and intuition, transform the information contained in the two-dimensional views to automatically produce a three-dimensional geometric solid.

10 In accordance with a preferred embodiment of the present invention, techniques are developed for defining an engine which has as its input two-dimensional drawing views, wherein the engine produces a three-dimensional geometric solid. The engine comprises, among other things, a set of rules for identifying particular elements within the various two-dimensional views, and for determining the useful relationships between elements
15 (e.g., arcs, lines, and the like) in one view *vis-a-vis* related elements in another view, to thereby ascertain the three-dimensional form defined by these elements and their corresponding relations to other elements.

 In accordance with a first embodiment of the present invention, a system is provided which permits an operator to interactively develop a plurality of two-dimensional views
20 relating to a three-dimensional object, whereupon the system automatically generates a three-dimensional geometric solid based on the operator's two-dimensional design drawing. In accordance with an alternate embodiment, the system is configured to retrieve (e.g. from memory, disc drive or the like) archival data relating to previously generated view sets, and to format the archival data as input to the rules engine associated
25 with the system to thereby generate a three-dimensional geometric solid for which two-dimensional drawings had previously been developed and stored.

 In accordance with a further aspect of the present invention, two-dimensional drawing data is automatically defined and organized into the appropriate view sets for processing by the system. In accordance with a still further aspect of the present invention,
30 relational constraints between elements within a view set as well as across view sets are automatically generated by the system. In accordance with a still further aspect of the

present invention, view sets derived from two dimensional drawing data are automatically ordered into plan view, front view, back view, and so on.

Brief Description of the Drawing Figures

5 The subject invention will hereinafter be described in conjunction with the appended drawing figures, wherein like numerals denote like elements, and:

Figure 1A is an illustration of the three basic Boolean operations as applied to solids;

Figure 1B is an illustration of the B-REP model of solid representation;

10 Figure 2 is a schematic block diagram of an exemplary embodiment of a system in accordance with the present invention;

Figure 3 is an example of a two-dimensional drawings corresponding to five views of a three-dimensional object;

Figure 4 is a flowchart showing an overview of an exemplary three-dimensional object generation method in accordance with the present invention;

15 Figures 5A and 5B are flowcharts setting forth an exemplary method for automatic definition and organization of view sets in accordance with the present invention;

Figure 5C depicts exemplary data structures showing the various records comprising the view cluster and view set structures;

20 Figures 6A and 6B are flowcharts setting forth an exemplary method of determining relational constraints within a view set.

Figures 7A and 7B are flowcharts setting forth an exemplary method of determining relational constraints across view sets.

Figures 8A and 8B are flowcharts setting forth an exemplary method for ordering view sets in accordance with the present invention;

25 Figure 8C depicts exemplary data structures showing the various records comprising the view set min-max data and extremes type structures;

Figures 9A, 9B, and 9C set forth an exemplary method of characterizing curve sets as closed, open, or disjoint, in accordance with the present invention;

30 Figure 9D depicts exemplary data structures showing the various records comprising the curve topology and closed curve set structures;

Figure 9E depicts exemplary data structures showing the various records comprising the open curve set and disjoint curve structures;

Figure 10 is a flowchart setting forth an exemplary method for establishing the main boundary for each view set;

5 Figure 11 is a flowchart setting forth an exemplary method for determining the plan view depth;

Figure 12A is a flowchart setting forth an exemplary method for determining the base solid type and creating said solid;

10 Figure 12B depicts exemplary data structures showing the various records comprising the curve identifier structures associated with atom definitions as well as the atom definition structure;

Figures 13A-J are flowcharts setting forth an exemplary method for processing disjoint curves;

15 Figure 13K depicts exemplary data structures showing the various records comprising the topological constraint identifier structure and the topological constraint set (TCS) structure;

Figures 13L and 13M are illustrations of an exemplary method used to position a curve set with respect to the base solid;

20 Figures 14A-J are flowcharts setting forth an exemplary method of processing closed curve sets; and

Figures 15A-R are flowcharts setting forth an exemplary method of processing open curve sets.

Detailed Description of Preferred Exemplary Embodiments

25 The present invention relates to a computer implemented technique for converting two-dimensional views of a three-dimensional object into a three-dimensional geometric solid. As such, the present invention may be implemented in any number of modalities, for example in the form of a software application for use on a CADD/CAM workstation or other personal computer. Similarly, the present invention may be implemented as a hardware system, for example in the context of a dedicated computer terminal.
30 Alternatively, the system may be implemented in any combination of a hardware, software,

or firmware environment, using any convenient user interface, for example conventional computer screen monitors, video monitors, or the like.

Referring now to Figure 2, an exemplary embodiment of the present invention suitably comprises a computer 200, a monitor 218, and an alphanumeric keyboard 220. Computer 200 suitably comprises a CPU 202, a power supply 204, a data communications bus 206, a memory 208, a disk controller 210, a diskette drive 212, a hard disk drive 214, and a video controller 216. Monitor 218 suitably comprises a conventional high resolution video monitor, for example, a monitor of the SVGA type commonly used with personal computers. Computer 200 suitably comprises a conventional desktop computer. Two-dimensional drawing data (described in further detail below) suitably resides and is manipulated within hard disk 212 and memory 208. Said data may be transferred to and from computer 200 through any number of means, such as by diskette, the World Wide Web, serial transfer, smartcard, scanning of hardcopy drawings, or through the interactive generation of data by an operator (e.g., a designer or draftsman) via a software application which runs on computer 200.

It will be appreciated that while various exemplary embodiments described herein comprise a traditional computer system operating in a sequential mode, the present invention may be advantageously utilized in a parallel processing environment.

A general description of the overall theory of operation of a preferred embodiment of the present invention will now be provided, followed by a more detailed discussion of the various steps, components, and logical modules for implementing the overall theory of operation.

Referring now to Figure 4, a flow chart of the general theory of operation of the present invention is provided.

Upon system start up (Step 1002), the various flags, registers, switches, and the like associated with a software implementation of the subject invention are initialized.

Upon identifying a particular two-dimensional data set for which a three-dimensional drawing is desired, either by interactively generating the data set, retrieving it from memory, reading the data set from a disk, smartcard, or other interface, the data set is then organized into a plurality of view sets (Step 1004).

More particularly and with momentary reference to Figure 3, a series of two-dimensional views are typically utilized to define a three-dimensional object. For purposes

of this discussion, each two-dimensional "view" is said to comprise a plurality of geometric "entities", which entities primarily comprise lines and arcs, but which also include splines, conics, and other features which make up a two-dimensional drawing. A particular two-dimensional view in this context refers to a planar projection of the three-dimensional object, for example a top (plan) view, front view, left view, right view, back view, bottom view, or oblique view. Moreover, a "view set" in the context of the present invention refers to the set of entities as seen in a particular view along with any applicable relations among those entities. For example, Figure 3 comprises five (5) view sets, corresponding to a plan view 302, a front view 304, a left end view 306, a bottom view 308, and an oblique view 310. It will be appreciated, however, that a drawing may comprise virtually any number of view sets to adequately define an object, depending on the complexity of the object. In this regard, it should be noted that the system may accept drawing data in a variety of formats. For example, the user may import data from the popular Initial Graphic Exchange Standard (IGES), or "STEP" (ISO 10303 Standard for the Exchange of Product Model Data) -- an international standard of increasing use in the industry -- as well as many lesser used standards.

Referring again to Step 1004 in Figure 4, the organization of view sets may be performed in a variety of ways. For example, in many instances this step may be performed manually. Such manual organization will typically require the user to select, via the system's graphical user interface, a set of drawing elements and designate those elements as comprising a separate, unique view set. Alternatively, the organization of view sets may be implicit in the data itself. More particularly, the data might be imported from a system in which the view set groupings are determined by the user during the design process itself (for example, the user may create the drawing in separate predefined windows corresponding to various views).

In a preferred embodiment, the organization of view sets is performed automatically. In this mode, the system determines which curves should be grouped together by examining, among other things, the relative proximity of the various geometric elements and chains of curves within the drawing. The manner in which the system performs this organization is explained in conjunction with Figures 5A and 5B below.

After organizing the two-dimensional drawing elements into view sets, the system then generates relational constraints among the various entities within a view set for each

view set (Step 1006). More particularly, a table is suitably generated for each view set which identifies and characterizes each and every entity (sometimes also referred to as an element) (e.g., line, arc, spline, etc.) within that view set.

5 More particularly, of the universe of geometric, spatial and other relationships which are possible among entities, the following subset of topological constraints (also referred to as relations or constraints) are advantageously employed in the context of a preferred embodiment of the present invention: coincident arcs, coincident endpoints, collinear endpoints, collinear lines, concentric arcs, endpoints on a curve, equal distance, equal length, equal radius, horizontal, parallel, perpendicular, symmetric, tangent, vertical, and
10 fixed (datum).

Coincident arcs may be defined as entities which share the same center, radius, start angle, and end angle within a particular planar view. Coincident endpoints are points which correspond (within some predefined tolerance) within a planar view. Collinear lines are entities (straight line segments) that lie on a single line. Concentric arcs are entities
15 (arcs) which have the same center within a planar view, but which may not have the same radius or start and end angles. Endpoints on a curve refer to elements with at least one endpoint lying on another element -- i.e., an element which terminates at its intersection with another element. Equal distance refers to a series of elements distributed at equal intervals along a line. Equal length refers to elements whose scalar lengths are equal.
20 Equal radius refers to elements (arcs and circles) whose scalar radius values are equal. Horizontal and vertical refer to lines which are parallel to the horizontal and vertical axes, respectively. "Endpoint on a curve" refers to the case where at least one of an element's endpoints lie on a curve -- i.e., when one curve terminates at another curve. Parallel, perpendicular, tangent, and symmetric are relations well known in the art. Note that not
25 just lines, but also pairs of curves, may be perpendicular at a point, and that two curves may also be parallel or tangent to each other.

Referring momentarily to Figure 3, it can be seen that a single element may be subject to many different relations with respect to other elements in a view set. For example, line 312 in plan view 302 shares coincident endpoints with both arc 321 and line
30 313. It is also parallel to a number of other lines in the view, including line 323 and line 328, as well being perpendicular to others, such as line 330 and line 314.

The particular manner in which the system analyzes the various elements within a view set (or among view sets) and establishes the relations among the various elements within a view set is described in greater detail below in conjunction with Figures 6A and 6B.

5 Referring now to Step 1008, the system then generates a second table, which advantageously exhibits structure similar to that described above in conjunction with Step 1006, setting forth all entities for all views and the relational constraints exhibited by each entity with respect to all other entities in all *other* view sets. Stated another way, the system then identifies the relationship between a particular entity and all other entities
10 contained in all other view sets, and performs this operation for every entity in every view set. For example, with reference to Figure 3, it can be seen that line 350 in front view 304 is tangent to circle 325 in plan view 302, and line 314 is collinear with line 342. These and all other combinations of relations between view sets are compiled by the system in Step 1008.

15 In this regard, it is worth noting that when a machinist studies a drawing containing a plurality of two-dimensional views, many of the relationships between an entity in one view and a related entity in another view may be ascertained by the machinist by inspecting the drawing. Indeed, it is the machinist's experience, intellect, and intuition which allows the machinist to create a three-dimensional mental picture of the object by
20 understanding the various two-dimensional views set forth in the drawing. In Step 1008, the system of the present invention imitates the mental processes performed by the machinist, and executes a rules engine for interpreting the relationships between entities in different views to facilitate the generation of a three-dimensional drawing, as discussed in greater detail below. The method of generating relational constraints across view sets
25 is described in greater detail below in conjunction with Figure 7.

After generating all applicable relations, the system then prioritizes the various view sets (Step 1010), for example by identifying the main or plan view, and thereafter defining the relationship of each of the other views with respect to the main view. The method employed by the system in defining view sets is discussed in conjunction with Figures 8A
30 and 8B below.

With reference to step 1012, the system then examines each view set, and divides the elements comprising a view set into a unique one of the following three categories: closed curve sets, open curve sets, and disjoint curves.

5 More particularly, a disjoint curve is a single element that shares no endpoints with any other element within the view set. With momentary reference to the sample drawing shown in Figure 3, lines 322 and 323 in plan view 302 are examples of disjoint curves, inasmuch as neither line shares an endpoint with any other curve in the view set.

10 An open curve set is a chain of curves wherein (1) each successive curve shares an endpoint with the previous curve, and (2) the chain of curves is not closed -- i.e., two elements within the chain each have one endpoint not coincident with any other element within the view set. Referring again to Figure 3, lines 327, 328, and 329 in plan view 302 represent an example of an open curve set.

15 A closed curve set is a chain of curves wherein (1) each successive curve shares an endpoint with the previous curve, and (2) the chain of curves is closed (there are no open endpoints). In Figure 3, the outer boundary defined by arc 321, line 312, line 313, line 314, arc 315, line 316, line 317, arc 318, line 319, and line 320 is an example of a closed curve set. The manner in which the system categorizes curve sets is discussed in greater detail with Figures 8A, 8B, and 8C below.

20 Having characterized each element of each view set as belonging to either a closed curve set, an open curve set, or as a disjoint curve, the system then establishes the main boundary for each view set (Step 1014). In particular and again referring to Figure 3, the main boundary for each view set may be described as the closed curve set having the largest area. The manner in which the system determines the main boundary for each view set is discussed in greater detail below in conjunction with Figure 10.

25 The system then isolates the main boundary for the plan view, and determines the view depth of the plan view main boundary (Step 1016). At a first level, this corresponds to identifying the gross shape of the main portion of the object under inspection, which may be thought of as corresponding to the main boundary as seen in the top view, and then sweeping that main boundary vertically to create a three-dimensional main boundary.
30 The extent to which this main boundary as seen in the plan view is swept vertically may be determined by inspecting the relationship between the plan view main boundary to the front view of that same main boundary as seen in the front view. An exemplary procedure

for determining the view depth of the main boundary is described below in conjunction with Figure 11.

Having determined the view depth of the main boundary of the plan view, the system then creates a base solid by performing, for example, an axial sweep (Step 1018).
5 In this regard, it should be appreciated that an axial sweep is but one of a plurality of techniques for generating a base solid, depending upon the particular configuration of the object under inspection. More particularly, there are three primary classes of solid generation methods: axial sweeps, rotational sweeps, and blended solids.

10 Axial sweeps are solids defined by the volume created by translation of a planar object axially through space. For example, a cylinder can be created through axial sweep of a circle. Rotational sweeps are solids defined by the volume created by rotation of a planar object through space around a line. Blended solids are solids defined by a series of curves, wherein straight, smooth, or rotational blends are used to transition along each successive curve.

15 Having created the base solid in Step 1018, the system then sets a "current view" flag to "one" (Step 1020); that is, the plan view is considered first.

Next, the set of disjoint curves within the current view are processed (Step 1022). This processing involves, in a broad sense, identifying disjoint curves in the current view then forming a solid atom to operate on the base solid using corresponding curves (e.g.,
20 collinear and tangent curves) present in other views. A detailed description of an exemplary method of processing disjoint curves is presented below in conjunction with Figures 13A through 13M.

Having processed the disjoint curves in Step 1022, the system then processes all closed curve sets within the current view (Step 1024). An exemplary method of
25 processing closed curve sets is presented below in conjunction with Figures 14A through 14J.

Finally, having processed all closed curves in Step 1024, the system then processes all open curve sets within the current view (Step 1026). An exemplary method of
30 processing open curve sets is presented below in conjunction with Figures 15A through 15P.

Having processed all disjoint, closed, and open curve sets, the system then increments the "current view" flag by one (Step 1030), then queries whether all views have been processed (Step 1030). If not, then Steps 1022, 1024, 1026, and 1028 are repeated until all views have been processed, at which time the system stops (Step 1032).

5 Finally, after processing all curve sets, the system takes the first two atoms (atom number one being the base solid generated previously in Step 1018) and, in accordance with methods well known in the art, applies the corresponding Boolean operation to the base solid. The resulting solid then becomes the new working base solid, and the system continues through the list of atoms until the finished geometric solid is created (Step
10 1032). After all views have been processed, a true three-dimensional geometric model of the target solid will reside in the system in such a form that it can be altered, supplemented, and exported for subsequent use. For example, the finished solid data may be exported to a variety of finite-element packages for further thermo-mechanical simulation. Similarly, in a production environment, the solid data may be exported to an
15 automated system suitable for fabrication of a finished product, for example, a numerical control system (NC).

Having presented an overview of the general theory of operation in conjunction with Figure 4, a more detailed description of the various steps will now be presented.

Referring now to Figure 5A, an exemplary method of automatically organizing view
20 sets (Step 1004 in Figure 4) will now be described. As noted above, the process of organizing view sets -- i.e., segmenting all drawing elements into groups corresponding to their respective views -- may suitably be (1) performed manually by an operator, (2) implicit in the data itself, or (3) performed automatically (or interactively) by the system.

In the automatic mode, the general flow presented in Figure 4 may be
25 advantageously modified. Specifically, as automatic organization requires categorization of curve sets as open, closed, and disjoint, the ordering procedure of Step 1012 as detailed below in conjunction with Figures 9A, 9B, and 9C is performed first. That is, if an operator chooses to allow the system to organize view sets, then Step 1012 is performed prior to step 1006; as such, Step 1012 need not be repeated between Steps 1010 and
30 1014.

Assuming that all curves have been categorized as open, closed, or disjoint, the system then continues as shown in Figure 5B. First, the closed curve sets are ordered by

decreasing area (Step 1126). Next, the center of gravity of each closed curve set is examined, and smaller closed curves with centers of gravity within larger closed curves are grouped with those larger closed curves (Step 1128). The phrase "center of gravity" in the context of the present invention suitably refers not to the precise computation required to compute the centroid of the region, but rather to the approximate coordinates calculated by computing the average of the X and Y extrema ranges; that is the center of gravity is approximated as (x,y) where $x=(XMAX-XMIN)/2$ and $y=(YMAX-YMIN)/2$. This approximation is adequate for the purpose of ordering and arranging view sets, although a more precise method may be used.

Suitable methods for computing the area of polygon in the context of the present invention include, for example, performing a planar integration of the area bounded by the curves or, when only a reasonable approximation is necessary, computing the area as:

$$\text{Area} = \frac{1}{2} ((x_1 * y_2 + x_2 * y_3 + \dots + x_n * y_1) - (y_1 * x_2 + y_2 * x_3 + \dots + y_n * x_1))$$

where (x_i, y_i) are coordinates of the vertices of a polygon defined by $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_n, y_n), (x_1, y_1)$.

After grouping closed curve sets as set forth above, open curve sets are then considered (Step 1130). Each open curve set is examined to determine whether it intersects or touches any closed curve. If so, then that open curve set is grouped with the closed curve. It should be noted that, in the context of a preferred embodiment of the invention, every open curve necessarily makes contact with at least one closed curve; that is, open curve sets should not exist in isolation within a well-formed two dimensional drawing of a three-dimensional object, since the resulting solid would likely be indeterminate and hence of limited utility.

Next, the system adds remaining closed curve sets to the sets thus far determined (Step 1132). Finally, each disjoint curve is added to the appropriate view set much in the same way open curve sets were dispositioned -- i.e., by looking for points of contact between the open curve set and other curves already categorized (Step 1134).

Having thus described a method of organizing view sets, an exemplary method of generating relational constraints within each view set will now be described. As a preliminary matter, an exemplary data structure used to characterize topological constraints will now be described in detail.

Referring now to Figure 13K, the system suitably generates a table 2088 which suitably comprises a data structure used to store information regarding relations (sometimes also referred to as topological constraints); in a preferred embodiment, a separate data structure 2088 may suitably be created for each view set. Alternatively, a
5 single data structure 2088 may be generated which includes the relational constraint information within a view set, for all view sets.

Data structure 2088 suitably comprises a first entry 2089 which suitably stores indicia of the number of topological constraint sets (nTCS) within the data structure 2088. Data structure 2088 may also comprise respective entries 2090(a), 2090(b) . . . 2090(n),
10 with each respective entry 2090 suitably comprising a unique identifier which identifies a unique, corresponding topological constraint set. In this regard, Figure 13K sets forth an exemplary schematic representation of a suitable topological constraint set 2091. That is, a plurality of constraint sets 2091 (not shown) are generated, with each one being identified by one of the identifiers 2090(a) -2090(n).

With continued reference to Figure 13K, an exemplary data structure 2091 suitably comprises a first field 2092 which identifies the type of topological constraint associated with that particular topological constraint set (TCS). As discussed above, a topological constraint set may comprise a set of entities which exhibit the relation "parallel", the relation "normal", the relation "collinear lines", and so on. Thus, in accordance with the
20 illustrated embodiment, entry 2092 in each data structure 2091 suitably comprises indicia of a unique one of the aforementioned relations which, in a preferred embodiment, comprise a total of seventeen (17) possible relations (parallel, normal, collinear lines, tangent, horizontal, vertical, symmetry, collinear end points, end points on a curve, concentric arcs, coincidental arcs, fixed (datum), point on two curves, coincident points,
25 equal radius arcs, equal length lines, and equal distance.

With continued reference to Figure 13K, data structure 2091 suitably comprises a second field 2093, which corresponds to the number of elements (sometimes also referred to as entities or simply "curves" herein) which share the topological constraint set forth in entry 2092 of the data structure. In this regard, if data structure 2092 relates to a
30 particular topological constraint for a view set, the number set forth in entry 2093 will correspond to the number of elements within that view set which share the relations set forth in entry 2092; alternatively, if data structure 2091 relates to an entire drawing

(corresponding to a plurality of view sets), then the number in entry 2093 suitably corresponds to the total number of curves in all of the view sets which share the relation defined in entry 2092.

5 Respective entries 2094(a), 2094(b) . . . 2094(n) of data structure 2091 suitably identify the particular elements (or curves) which share the relation defined in entry 2092. It will be appreciated that the number of entries should correspond to the numbers set forth in entry 2093.

10 With continued reference to Figures 3 and 13K, a specific example of a topological constraint set will now be described to further clarify the nature of the data set forth in data structure 2089. Let us select the topological constraint "tangent" for this example, such that some convenient indicia of the "tangent" relation would be set forth in entry 2092. Let us further assume that the data structure 2091 relates to all of the view sets shown in Figure 3, as opposed to relating to only a single view set within Figure 3. In that case, the number of elements shown in Figure 3 which share the "tangent" relation with
15 the other elements to be listed would be set forth in entry 2093. Although a number of elements within the various view sets shown in Figure 3 satisfy the "tangent" relation, the following specific examples illustrate the nature of the data in data structure 2091, while at the same time highlight the manner in which the present invention emulates the experience and intellect of a machinist by embodying some of the essential techniques
20 mentally employed by machinists into executable codes.

 With reference to element 324 in plan view 302 of Figure 3, it can be seen that line segment 348 is tangent to curve 324. Thus, an identifier corresponding to curve 324 might suitably be set forth in first curve identifier field 2094(a), while an identifier associated with curve 348 might suitably be set forth in second curve identifier entry
25 2094(b). Similarly, it can also be seen that line 350 (shown in front view 304 of Figure 3) is tangent to circle 325 (shown in plan view 302); hence, respective identifiers for each of curve 350 and 325 would also be suitably set forth in respective curve identifier entries within data structure 2091.

30 Moreover, a separate data structure 2091 is advantageously constructed for each of the possible topological constraints set forth in field 2092, although it will be understood that only a subset of the various possible constraints will exist for any particular drawing.

Having thus described exemplary data structures used to characterize topological constraints, an exemplary method generating said constraints will now be described. Referring now to Figures 6A and 6B, the system first builds a list of curve identifiers (CURVLIST) for the first view set (Step 1202). Next, the first curve identifier (ACTCURVID) is placed into the active curve register (Step 1208).

In Step 1212, the system queries whether ACTCURVID is a line. If so, the system proceeds to Step 1214; if not, the system proceeds to Step 1218, described below.

In the case where ACTCURVID is a line ("yes" branch from Step 1212), the system in Steps 1214 and 1216 loops through all possible constraint types that can apply to a line (suitably comprising parallel, collinear, horizontal, endpoint-on-a-curve, normal, tangent, vertical, and equal length) and when a constraint type is satisfied, checks whether a TCS exists for it. If such a TCS exists, the system adds ACTCURVID to the existing TCS; if not, the system starts a new TCS of the type being processed. After thus creating or appending to a TCS, the system queries whether All curves have been processed (Step 1210). If so, the system continues at Step 1206, described below; if not, the system selects the next curve identifier in CURVLIST, places it in the active curve register, and continues as before at Step 1212.

In the case where the active curve is not a line ("no" branch in Step 1212), the system queries whether the active curve is an arc or circle (Step 1218). If not, the system proceeds to Step 1224, described below. Referring now to Figure 6B, if the active curve is a spline, the system in Steps 1220 and 1222 loops through all possible constraint types that can apply to an arc (suitably comprising concentric arc, equal radius arc, coincident arc, tangent, endpoints-on-a-curve) and when a constraint type is satisfied, checks whether a TCS exists for it. If such a TCS exists, the system adds ACTCURVID to the existing TCS; if not, the system starts a new TCS of the type being processed. After thus creating or appending to a TCS, the system queries whether All curves have been processed (Step 1210). If so, the system continues at Step 1206, described below; if not, the system selects the next curve identifier in CURVLIST, places it in the active curve register, and continues as before at Step 1212.

If at Step 1218 the system did not find an arc or circle, the system comprehends in an exemplary embodiment that the active curve must be either a conic or a spline. The system in Steps 1224 and 1226 loops through all possible constraint types that can apply

to a conic or spline (suitably comprising horizontal, vertical, parallel ellipses, coincident ellipses, and endpoint-on-a-curve) and when a constraint type is satisfied, checks whether a TCS exists for it. If such a TCS exists, the system adds ACTCURVID to the existing TCS; if not, the system starts a new TCS of the type being processed. After thus creating
5 or appending to a TCS, the system queries whether All curves have been processed (Step 1210). If so, the system continues at Step 1206, described below; if not, the system selects the next curve identifier in CURVLIST, places it in the active curve register, and continues as before at Step 1212.

10 In Step 1206, after the system has processed all curves and identified their respective topological constraint sets, the system queries whether all view sets have been processed. If not, the system loops back to Step 1202, increments the view, and continues as before with Step 1208. If all view sets have been processed, the system exits the subroutine (Step 1204).

15 Having thus generated all relational constraints within each view set, the system then similarly generates all constraints across view sets (Step 1008 in Figure 4). Referring now to Figure 7, an exemplary method of doing so will now be described. First, in Step 1302, the system builds a list of curve identifications (CURVLIST) for the first view set (ACTIVEVUSET). Next, the system builds a list of curve identifications (2NDCURVLIST) for the first view set after ACTIVVUSET (Step 1306). The first curve
20 identifier from CURVLIST (ACTCURVID) is then placed in the active curve register (Step 1310).

In Steps 1314 and 1316, the system loops through all possible constraint types that can apply between view sets (suitably comprising collinear lines, horizontal, endpoints-on-a-curve, tangent, vertical, and equal length) until all constraint types have been processed.
25 When a constraint type is satisfied, the system checks whether a TCS exists for it. If so, the ACTCURVID is placed in the existing TCS; if not, a new TCS is created and ACTCURVID is placed in the this new TCS.

Having generated all constraints between the instant pair of view sets, the system then queries whether all curves in CURVLIST have been processed (Step 1312). If so,
30 the system proceeds to Step 1308, described below; if not, the system selects then next curve identifier from CURVLIST, places it into the active curve register, then continues at Step 1314 as before.

In the case where all curve in a particular CURVLIST have been processed, the system continues on to Step 1308 and determines whether all view sets after ACTIVVUSET have been processed. If not, the system in Step 1306 builds another list of curve identifications for the next view set after ACTIVVUSET and continues at Step 1310. When all view sets after ACTIVVUSET have been processed ("yes" branch in Step 1308) the system exists the subroutine (Step 1304).

An exemplary procedure for ordering view sets (Step 1010) will now be described in detail. In a broad sense, this step involves examining the location of each view set with respect to the plan view and, given common drafting, design, and engineering conventions, labeling and ordering each of these view sets as front, back, and so on. For example, with momentary reference to Figure 3, it can be seen that the sample drawing comprises five separate views. Even if the views were not labeled as front, left end, bottom, and so on, a competent machinist or draftsman would intuitively ascertain that the view set immediately to the left of the plan view (left end view 306) corresponds to the left end view, and that the view set immediately below the plan view (front view 304) corresponds to the front view of the object; that is, the positioning of views is standardized within the drafting community in order to facilitate graphical communication -- for example, through the American National Standards Institute drafting standards (ANSI). Nevertheless, the process by which this categorization may be automated is non-trivial, and is the subject of the following discussion.

Referring now to Figures 8A and 8B, the system first determines the maximum X and Y coordinate values for data within each view set (Step 1402). In this regard, the system assumes that when the view sets were previously defined and organized (Step 1004), the plan view was designated as the first view set. This is a prerequisite for correct interpretation of the remaining view sets. These values are labeled XMIN, XMAX, YMIN, and YMAX. The center of gravity of the plan view set is then determined (Step 1404), and this data is labeled CGX and CGY. Referring momentarily to Figure 8C, the view set min-max data structures 1470 and 1472 are used to store values generated during Step 1402. Structure 1470 suitably comprises extremum and center of gravity values for each view set. Structure 1472 suitably comprises an identification of the corresponding curve and extremum types for each extremum in the view set. Structure 1474 sets forth a suitable list of five possible extremum types (i.e., locations on a curve where an extremes

might occur), comprising: first endpoint of curve, second endpoint of curve, horizontal curve, vertical curve, and within an arc segment.

Referring again to Figures 8A and 8B, the system then initializes to the start of the view set list (Step 1406). The X and Y coordinates of the center of gravity of the current view set is determined, and the variables CGCX and CGCY are assigned the X and Y values respectively (Step 1408). Next, the distance of the current view set from the plan view is determined (Step 1410). Specifically, the following values are computed:

$$DX = CGCX - CGX$$

$$DDX = \text{ABS}(DX)$$

$$10 \quad DY = CGCY - CGY$$

$$DDY = \text{ABS}(DY)$$

The system then determines whether $DX=0.0$ and $DY=0.0$ (Step 1412). If so, then the current view is marked as the plan view (view 1), and is marked as 'USED' (Step 1414), after which the system increments to the next view set (Step 1440) and checks whether all view sets have been processed (Step 1442). If all view sets are processed, then the ordering operation is complete. If not, the system continues by looping back to step 1408. If, at step 1412, it was determined that the current view set did not correspond to the plan view, the system progresses to step 1416.

In Step 1416, as well as Steps 1420, 1424, 1428, 1432, and 1436, the system similarly checks the values of DX , DY , DDX , and DDY against a particular set of criteria for each view in order to characterize the current view. More particularly, the current view is designated the front view (view 2) when $DX=0.0$ and $DDY > 3.0*DDX$ (Step 1416), the right end view (view 3) when $DX < 0.0$ and $DDX > 8.0*DDY$ (Step 1420), the left end view (view 4) when $DX > 8.0*DDY$ (Step 1424), the back view (view 5) when $DY < 0.0$ and $DDY > 8.0*DDX$ (Step 1428), and the bottom view (view 6) when $DDX > 12.0*DDY$ (Step 1432) or $DDY > 12.0*DDX$ (Step 1436). If none of these criteria are met, the system then checks in Steps 1444, 1448, 1452, 1456, and 1460 whether the current view is an oblique view.

In accordance with steps 1440 and 1442, this ordering procedure continues until all view sets are processed and the system has identified and numbered each view.

An exemplary method for parsing entities into open curve sets, closed curve sets, and disjointed curves will now be described in detail. Referring to Figures 9D and 9E, the

system suitably uses four data structures for organizing curve sets: curve topology structure 900, closed curve set structure 920, open curve set structure 940, and disjoint curve structure. Curve topology structure 900 comprises data related to the number and identification of closed, open, and disjoint curves, and suitably comprises a closed curve set count 902, a first structure identifier 904, an open curve set count 906, a second structure identifier 908, a disjoint curve count 910, and a third structure identifier 912. This high level structure is suitably used, therefore, as simply a pointer to the various curve sets ordered previously in Step 1012.

Closed curve set structure 920 suitably comprises data specifying the contents of each closed curve set, and suitably includes, for each closed curve set, a curve count 922 (the number of curves within the set), a first curve identifier 924, a second curve identifier 926, and subsequent curve identifiers for each curve within the curve set. Open curve set structure 940 suitably comprises data specifying the contents of each open curve set, and suitably includes, for each open curve set, a curve count 942, a first curve identifier 944, a second curve identifier 946, and subsequent curve identifiers for each curve within the curve set.

Disjoint curve structure 960 comprises a list of all individual disjoint curves, and suitably includes a curve count 962 and curve identifier 964 for each of the disjoint curves.

Referring now to Figures 9A, 9B, and 9C, a preferred method used to characterize curves will be described. First, as shown in Step 1502, an "endpoint structure" table of endpoints is generated for all curves. This data structure (1504) suitably comprises four scalar endpoint values for each curve: X1, Y1, X2, and Y2. Next, an "identification structure" table is created (Step 1506) which suitably lists three indicia for each curve: a curve identifier, the view set location, and a flag indicating whether the curve is "available" (i.e., whether it has already been included in a curve set).

After creation of the end point and identification structures, the system queries whether any curves are "available" (Step 1510). If not, the ordering procedure is complete. If at least one curve is designated as "available," the system continues to Step 1512.

In Step 1512, the next "available" curve is chosen as the start of the current curve set and is marked as "not available." If no remaining curves are available (Step 1514), then the last curve is added to disjoint curves structure 960. If, at Step 1514, the system

determines that additional curves are available, the next available curve is designated as the active test curve (ATC) (Step 1518).

5 The system then determines in Step 1520 whether the first end of ATC matches the first end of the current curve set. If not, the system proceeds to Step 1524. If so, then the second end of ATC is used as the first end of the current curve set (Step 1522), the ATC is added to the start of the curve set (Step 1528), and the system proceeds to Step 1546 (detailed below).

10 In Step 1524, the system determines whether the second end of ATC matches the first end of the current curve set. If not, then the system proceeds to Step 1530. If so, then the first end of ATC is used as the first end of the current curve set (Step 1526), the ATC is added to the start of the curve set (Step 1528), and the system proceeds to Step 1546 (detailed below).

15 In Step 1530, the system determines whether the first end of ATC matches the second end of the current curve set. If not, then the system proceeds to Step 1534. If so, then the second end of ATC is used as the second end of the current curve set (Step 1532), the ATC is added to the end of the current curve set (Step 1540), and the system proceeds to Step 1546.

20 In Step 1534, the system determines whether the second end of ATC matches the second end of the current curve set. If not, the system proceeds to Step 1538. If so, then the first end of ATC is used as the second end of the current curve set, the ATC is added to the end of the current curve set (Step 1540), and the system proceeds to Step 1546.

25 In Step 1540, the system determines whether the curve set is closed (that is, whether the first end is coincident with the second end). If it closed, then the curve set is added to closed curve set structure 920 (Step 1552). If it is not closed, then the system proceeds to Step 1538.

30 In Step 1538, the system determines whether any non-checked curves are "available." If so, then the system loops back to Step 1518. If no more non-checked curves are "available," then the system looks at the number of curves within the current curve set (Step 1544). If there is more than one curve in the current curve set, then the curve set is added to open curve set structure 940 (Step 1550) and the procedure is complete. If there is only one curve within the current curve set, then the last curve is added to disjoint curve structure 960 (Step 1542) and the system proceeds to Step 1548.

In Step 1548, the system determines whether any more curves are "available." If more curves are available, then the system loops back to Step 1518. If not, then the procedure is complete.

5 Referring now to Figure 10, an exemplary method for establishing a main boundary for each view set in accordance with a preferred embodiment of the present invention will now be described.

The system first sets the current view to "one" (step 1602), which corresponds to initially interrogating the plan view, inasmuch as the plan view was ordered as the "number one" view in step 1010.

10 Having isolated the then current view (at this stage of the process, the plan view), the system orders the closed curve sets associated with the then current view in order of decreasing area within the curve set (step 1604). That is, at step 1604 the system identifies the closed curve sets associated with the then current view, and assigns order number 1 to the closed curve set having the largest area, order number 2 to the curve set
15 having the next greatest area (if one exists), and so on.

More particularly, the system suitably interrogates closed curve set structure 920 (step 1552), and selects the first closed curve set in the data structure. The system then computes the area within the first closed curve set or, alternatively, the system simply
20 interrogates a field within data structure 920 which sets forth the area for that closed curve set if the area has previously been ascertained, for example, in connection with the processes described in connection with Figure 9.

The system then interrogates the next closed curve set within data structure 920 (if any), and similarly computes the area within that closed curved set, and so on. When the area has been calculated for each closed curve set within a particular view, they are simply
25 ordered in terms of magnitude, with the curve set having the greatest area identified as the first closed curve set for the then current view set. Having ordered the closed curve sets by area for the then current view set, the system then increments the current view by one (Step 1606); the system then checks to see if the closed curve sets have been ordered for all view sets (Step 1608). If closed curve sets for all view sets have not yet been ordered
30 ("no" branch from step 1608), the system returns to step 1604 to effect the ordering, by area, of the closed curve sets corresponding to the then current view.

Returning to step 1608, if the closed curve sets have been ordered for all views (corresponding to the "yes" branch from step 1608), the system determines the main boundary for each view set (step 1610). Specifically, the system selects the highest ordered closed curve set (as determined in step 1604) which corresponds to the closed
5 curve set having the greatest area within a particular view set, and defines that highest area closed curve set as the main boundary for that particular view. The system similarly identifies the main boundary for every view set.

In this regard, it can be seen with reference to Figure 3 that the plan view 302 comprises a plurality of closed curve sets. In particular, respective entities 324 and 325
10 each comprise a closed curve set, even though they each comprise a single element (namely, a circle). In addition, closed curve set 330 is defined by a plurality of adjacent curves which, for clarity, are not separately labeled. As can be seen by casual visual inspection of plan view 302, the closed curve set having the greatest area is that curve set
15 which comprises the contiguous entities 320, 321, 312, 313, 314, 315, 316, 317, 318, and 319. As alluded to above, while a skilled machinist can readily and intuitively identify the "main boundary" associated with plan view 302, prior art systems have simply not appreciated the desirability of implementing this process in a software-based system in the context of a system for generating a 3-dimensional rendering of a solid object from a plurality of 2-dimensional views of the object. Indeed, one of the key elements of the
20 present invention surrounds the present inventor's appreciation of the need to identify and define the important attributes of a machinist's skill and experience, and to replicate these processes in machine executable code.

Referring now to Figure 11, an exemplary method for determining the plan view depth (step 1016) will now be described.

25 Recall from Figure 10 that the system of the present invention determined the main boundary for each view set, wherein the main boundary for a particular view set corresponds to the closed curve set having the greatest area within a view set. In the context of Figure 11, the system focuses on the main boundary of the plan view.

30 With regard to Figure 11, it will be appreciated that an important technique employed in context of the present invention surrounds the notion of identifying a "main body" or "main solid," which essentially performs the function of identifying a baseline solid which grossly approximates, at least in terms of aggregate size, the final 3-

dimensional object under inspection. By analogy, this can be thought of as the process undertaken by a sculptor when creating a detailed statue, figurine, or the like. The sculptor starts with a large block or other form which is initially devoid of relevant detail, but which nonetheless functions as a gross approximation of the finished artwork. Just
5 as the sculptor removes a desired amount of material from specific locations of the block to progressively arrive at a desired figure, the system in accordance with the present invention employs boolean operations to successively remove predetermined shapes and volumes from a gross main body to ultimately arrive at the final 3-dimensional geometric solid.

10 Accordingly, although the sculptor's starting block of material bears little resemblance to the finished statue, a sculptor must nonetheless start with something he can work with. Analogously, the system of the present invention strives to identify a gross main body which it then uses to incrementally redefine into the ultimate finished 3-dimensional drawing. The procedure described below in connection with Figure 11 relates
15 to an exemplary technique for conveniently defining a gross 3-dimensional structure as a starting point for future definition. Succinctly, the system isolates a main boundary from the plan view (Figure 10), and then sweeps or otherwise manipulates the plan view main boundary to produce a 3-dimensional base solid (as discussed, for example, in connection with step 1018 of Figure 4). However, in order to determine the base solid from the plan
20 view main boundary, the system needs to first ascertain the extent of the axial sweep (or other extrapolation of the main boundary into 3-dimensional space); the determination of the extent to which the plan view main boundary should be vertically swept to create a base solid is the focus of Figure 11.

The system inspects the coordinates of the plan view main boundary, and determines
25 the coordinate extremis associated with that main boundary (step 1602). More particularly, the system inspects the X- and Y-coordinate values associated with the main boundary closed curve set, and in so doing, identifies and captures the point on the main boundary having the smallest X-coordinate value (XMIN), the point on the main boundary curve having the smallest Y-coordinate value (YMIN), the point on the curve having the
30 largest X-coordinate value (XMAX), and the point on the main boundary curve having the largest Y-coordinate value (YMAX).

Having determined the plan view main boundary extremes, the system then inspects the front view to ascertain whether the main boundary associated with the front view shares the same XMIN and XMAX values previously determined in step 1802. If the XMIN and XMAX values associated with the front view main boundary are the same as the XMIN and XMAX values associated with the top view main boundary ("yes" branch of step 1806), the system concludes that an appropriate depth associated with the plan view main boundary may be advantageously defined by the YMAX and YMIN values associated with the front view (Step 1806). Specifically, in accordance with the illustrated embodiment, the system selects the scaler value (YMAX - YMIN) as the "depth" of the plan view main boundary.

If, on the other hand, the plan view XMIN and XMAX values do not correspond with the XMIN and XMAX values of the front view ("no" branch from step 1806), then the system concludes that it must look elsewhere, i.e., it must look to a view other than the front view to determine the appropriate "depth" of the plan view main boundary. Thus, the system next compares the plan view XMIN and XMAX values with the XMIN and XMAX values of the main boundary associated with the rear view (step 1810). If they are the same ("yes" branch from step 1812), the "depth" of the plan view main boundary is advantageously defined as the scaler value (YMAX - YMIN) taken from the rear view (step 1814). If the plan view X-coordinate extremis do not coincide with the rear view X-coordinate extremis ("no" branch from step 1812), the system again concludes that it must continue searching for an appropriate yardstick for measuring the appropriate "depth" of the plan view main boundary.

In particular, the system then compares the plan view XMIN and XMAX with the XMIN and XMAX values associated with the right view (step 1816), and if correspondence is ascertained ("yes" branch from step 1818), the system defines the plan view main boundary depth as the magnitude of the difference between XMAX and XMIN associated with the rear view. If the X-coordinate extremis of the plan view do not correspond with the X-coordinate extremis of the right view ("no" branch from step 1818), the system then looks for a match in the left view (step 1822). If a match is found ("yes" branch from step 1824), the system defines the depth of the plan view main boundary as the difference between the XMAX and the XMIN value of the left view (step 1826). If no match is found ("no" branch from step 1824), meaning that the plan view

main boundary X-coordinate extremis did not match up with the X-coordinate extremis in either the front view, rear view, right view, or left view, then the system generates an error signal. This could mean that one or more of the entities in one or more of the view sets was either improperly entered, improperly characterized, or corrupted. Intuitively, this means that the three views which are orthogonal to the plan view (front, right, and left) do not include a main boundary which, when viewed in conjunction with the plan view main boundary, result in a solid object.

Referring again to Figure 4, recall that at Step 1016, the system determines the view depth of the plan view main boundary. Having determined the view depth of the main boundary of the plan view, the system then creates a base solid (Step 1018). As also briefly discussed above, a suitable base solid may be generated using one of several techniques, depending on the principal geometric characteristics of the three dimensional solid under inspection. For example, the system may employ, *inter alia*, the following solid generation methods: axial sweeps, rotational sweeps, blended solids, and curve driven solids. An exemplary methodology for determining which solid generation technique is most appropriate for the particular three dimensional object under inspection will now be described.

More particularly and referring now to Figure 12A, the various criteria for selecting an optimum (or at least appropriate) solid generating technique are set forth. The system first isolates the top view (Step 1802), inasmuch as most of the information necessary to make an initial determination as to the appropriate solid generation model to be employed can be gleaned from an inspection of the curves set forth in the top view.

The system then inspects the curves associated with the top view to determine whether all of the curves in the top view comprise one or more closed curve sets (Step 1806). If all of the curves in the top view are not closed curve sets ("no" branch from Step 1806), the system concludes that neither a rotational sweep nor a smooth blended object is under inspection, and thus employs an axial sweep (Step 1818), for example as discussed above in connection with Step 1018 in the context of Figure 4.

In this regard, it should be noted that for most solids generated by the subject system, an axial sweep methodology will likely be the most appropriate solid generation technique. However, the system recognizes that certain special cases may be encountered from time to time, wherein a rotational sweep or smooth blended technique are

appropriate. The present inventor has determined that the rotational sweep and smooth blended techniques are only possible if, among other things, every curve in the top view corresponds to a closed curve set. Thus, if at least one curve associated with the top view is not part of a closed curve set, the system can immediately conclude that an axial sweep
5 (which may also be thought of as the default condition for solid generation) will be employed.

With continued reference to Step 1806 of Figure 12A, if the system determined that all curves in the top view correspond to closed curve sets ("yes" branch from Step 1806), the system undertakes further analysis to determine one of the special cases for rotational
10 sweep or smooth blended solid generation techniques may be employed.

More particularly, upon determining that all curves in the top view set correspond to closed curve sets, the system may suitably be configured to finally determine whether a rotational sweep technique may be employed. In this regard, a rotational sweep technique for generating a base solid is particularly appropriate wherein the solid object
15 under inspection may be described as having a vertical axis, and further wherein the object is symmetric about any vertical plane which includes the vertical axis of the object. One can visualize such an object as, for example, a simple rounded bowl, flower vase, or the like, which, in top view, comprises a series of concentric circles. Thus, one technique employed in the context of the present invention for determining whether the object under
20 inspection satisfies these criteria is to ascertain whether all of the closed curves in the top view comprise concentric circles (Step 1810). If all of the closed curves in the top view are not concentric circles ("no" branch from Step 1810), the system concludes that the rotational sweep technique is not appropriate, and goes on to determine whether a smooth blended solid generation technique would be appropriate.

25 If, on the other hand, all closed curves in the top view are concentric circles ("yes" branch from Step 1810), the system determines whether all other rotational sweep criteria are satisfied (Step 1812). If all other rotational sweep criteria are not satisfied ("no" branch from Step 1812), the system proceeds to Step 1814 to determine whether a smooth blended solid generation technique may be employed. If, on the other hand, all
30 other rotational sweep criteria are satisfied ("yes" branch from Step 1812), the system employs a rotational sweep solid generation technique (Step 1808).

More particularly and with continued reference to Step 1810, having determined that all top curves constitute closed curve sets and that the closed curve sets are all concentric circles, the system determines whether both of the following criteria are satisfied: (i) do the concentric circles constitute circular arrays of closed curve sets
5 concentric with the main boundaries; and (ii) is the additional view (i.e., in addition to the top view) axi-symmetric about its center (i.e., if the additional view is a front or back view, it will be axi-symmetric about the vertical axis; if the additional view is a left or right view, it is axi-symmetric about its horizontal axis). In the illustrated embodiment, the system employs a rotational sweep (Step 1808) for generating the base solid only if
10 the foregoing criteria are satisfied as part of the analysis performed in Step 1812. If both of these criteria are not satisfied, the system determines whether the smooth blended criteria are satisfied (Step 1814).

More particularly, in Step 1814, the system determines if the additional view (i.e., the view shown in the drawing in addition to the top view) is a front view; if it is not, the
15 system performs an axial sweep. If the additional view is a front view, the system determines whether the front view consists only of horizontal lines whose ends would coincide with the extensions of vertical tangents to the extrema of each top view closed curve set. If this criteria is satisfied in Step 1814, the system concludes that a smooth blended solid generation technique is appropriate (Step 1816).

Returning briefly to Step 1814, if the system determines that the additional view is either not a front view, or if the system determines that the additional view is a front view, but that it does not consist only of horizontal lines whose ends would coincide with the extensions of vertical tangents to the extrema of each top view closed curve set ("no"
20 branch from Step 1814), the system concludes that neither the rotational sweep nor the smooth blended solid generation techniques are appropriate, and thus proceeds to perform an axial sweep (Step 1818).
25

Referring again to Step 1808, a base solid may be generated using the rotational sweep technique in accordance with one aspect of the present invention by ascertaining the profile of the three dimensional object and rotating that profile about the axis of
30 symmetry. This may suitably be accomplished through the use of commonly known methods, for example, by computing the locus of points resulting from a curve rotated in arbitrarily small increments about a straight line.

Returning now to Step 1812, a base solid may be generated using the smooth blended technique in accordance with the present invention, for example, by using a series of closed curve sets, wherein either straight or smooth blends are used to transition between each successive curve.

5 Referring now to Figure 13A, an exemplary method for processing disjoint curves will now be described. In the general sense, this process involves carving away at the base solid by closing off each disjoint curve, rotating or otherwise manipulating the resulting curve set into place with respect to the base solid, then applying solid generation techniques to the curve set in order to form the appropriate solid atom.

10 The number of disjoint curves in the current view is first placed in the active register (Step 1902). Note that this number corresponds to first record 962 within disjoint curve structure 960 for the current view. The system then extracts the first curve identifier from this structure (Step 1904) and categorizes the orientation of the corresponding curve as either horizontal, vertical, or oblique. For curves that are lines, horizontal means parallel to the X-axis, and vertical means parallel to the Y-axis. Oblique lines are simply those
15 lines which are neither vertical nor horizontal. For curves which are not lines -- i.e., arcs, conics, and splines -- the system operates as though an implied line segment exists between the curve's endpoints and then categorizes the orientation of the curve according to the orientation of this implied line segment.

20 The system then examines the disjoint curve (DC) to determine whether it is vertical (Step 1908). If not, then the system proceeds to Step 1924. If so, then the system further queries whether the curve is a non-line (i.e., a member of the group suitably comprising arcs, conics, and splines). For lines, the system proceeds to Step 1912 (Process Vertical Disjoint Line), which is discussed below in conjunction with Figures 13B, 13C, 13D, 13L,
25 and 13M. For non-lines, the system proceeds to Step 1914 (Process Vertical Disjoint Curve), discussed below in conjunction with Figure 13E.

If previously at Step 1908 the system did not find DC to be vertical, then the system proceeds to Step 1914, and queries whether DC is horizontal. If not, then the system proceeds to Step 1922; if so, then the system branches according to whether the curve is
30 a line or non-line (Step 1926). For lines, the system proceeds to Step 1928 (Process Horizontal Disjoint Lines), which is discussed below in conjunction with Figure 13F. Non-

lines are processed according to Step 1930 (Process Horizontal Disjoint Curve), which is discussed below in conjunction with Figure 13I.

For curves that are oblique (neither vertical nor horizontal), the system proceeds to Step 1922 (Process Oblique Disjoint Line), which will be discussed in further detail below
5 in conjunction with Figure 13J.

After the current disjoint curve DC has been categorized and processed through the appropriate subroutine (Steps 1912, 1914, 1928, 1930, and 1922), the system adds the resulting axial sweep atom to the atom definitions structure (Step 1916). In this regard, and with reference to Figure 12B, atom definitions structure 1822 comprises records
10 which identify the solid atoms produced through processing of the disjoint, closed, and open curves. This structure suitably includes a count of the number of atoms 1824 and, for each atom, the following eight records 1826(a)-(n): a curve set identifier, a Boolean key identifying how the atom should operate on the main solid (1=union, 2=difference, 3=intersection), a work view number designated which view was utilized to create the
15 atom, solid type (1=axial sweep, 2=rotational sweep, 3=smooth blended), the atom ZTOP, the atom ZBOTTOM, the ZTOP blend radius, and ZBOTTOM blend radius. ZTOP and ZBOTTOM, as will be seen shortly, refer to the extent of sweep that the solid will undergo during atom creation.

If not all disjoint curves within the view have been processed (Step 1918), the next
20 disjoint curve is selected and, returning again to Step 1906, the procedure is repeated until the list of disjoint curves within the view set has been exhausted.

Having thus described the main procedural loop used to process all disjoint curves within a view set, we now turn to the five logical modules used to create the solid atoms which will function to effectively "build upon" or "carve away at" the main solid. In this
25 regard, Figures 13L and 13M will be referred to intermittently in order to impart an intuitive understanding of the system's operation.

Referring now to the flowchart in Figure 13B, an exemplary method of processing vertical disjoint lines will now be described in detail (Step 1912 in Figure 4). First, the topological constraint sets (TCSs) are examined until the identifier for the active disjoint
30 line (ADL) is found (Steps 1930, 1932, and 1934). In a preferred embodiment, the system considers in its search only a subset of TCS relation. More particularly, the present inventor has determined that three primary relations may be advantageously employed in

characterizing the vast majority of objects likely to be encountered in practice; that is, tangent, collinear line, and endpoint on a curve.

If all TCSs have been explored, and the ADL identifier has not been found, then the system proceeds to Step 1936. After setting the ERROR register to "NO CONSTRAINT
5 FOUND", the system exits the subroutine (Step 1942).

Referring now to Figure 13C, if the ADL was found previously in Step 1932, then the system proceeds to Step 1943 and locates another curve in the same TCS which is *not* in the current view. This new view is designated as the "work view." Note that since the present subroutine comprehends only vertical lines, the work view will necessarily be
10 either the front or back view; that is, the only views that may comprise elements which bear the relationship of tangent or collinear with respect to ADL are the front or back views. The collinear or tangent line identified thereby is designated as MATCHCURV (Step 1945). For example, and with momentarily reference to Figure 13L, note that if plan view vertical line 2090 is the ADL, then collinear line 2091 fulfills the requirements for
15 MATCHCURV: it is collinear and exists within a different view (front).

Having identified MATCHCURV, the system then determines whether MATCHCURV is an element in the work view main boundary (Step 1946) or an element within an "interior" closed curve set (Step 1947). "Interior" in this context means a curve set which describes an area which lies within the larger main boundary. With momentary
20 reference to Figure 13L, MATCHCURV 2091 is an example of an element in the work view main boundary.

If MATCHCURV is part of the work view main boundary, then the system proceeds to Step 1950. If MATCHCURV is part of an interior closed curve set, the system proceeds to Step 1953. If MATCHCURV does not fit within either category, then
25 the ERROR register is set to 'No Atom Defined' (Step 1948), and the subroutine ends (Step 1949).

If MATCHCURV is in an interior closed curve set, this set is moved into the active curve set (ACS) and the system then proceeds to Step 1953 described below.

Consider now the work view main boundary scenario (Step 1950). Here, the system
30 starts at MATCHCURV and progresses forward and (if necessary) backward along the main boundary closed curve set until the resulting chain of curves encounters exactly two from the set XMIN, YMIN, XMAX, and YMAX. In other words, the system identifies

a short chain of elements whose outermost members terminate at the boundary of the main outline. Thus, for example, the region 2092 in Figure 13L is bounded in part by a set of three contiguous elements (two lines and an arc). This chain of elements intersects YMAX and XMIN in the front view. The resulting boundary curves then are ordered such that the first Extremum touching curve is listed 1st, and the last Extremum touching curve is listed last within the active curve set. In this regard, note that the main boundary is a closed curve set, and a basic feature of closed curve sets is that they can be termed "SCROC"s ; that is, such a set is Simple (does not cross itself), Closed (the start point is the end point), Rectifyable (has a finite length), Orientable ("inside" is always on left while moving along curves counter clockwise), and a Curve (or set of curves).

In Steps 1951 through 1952, one or more "phantom curves" are added to the chain of boundary curves in order to close it off -- i.e., to create a closed curve set which will be used to define the solid atom. Specifically, the active curve set (ACS) is closed using the rules set forth below in Table 1.

TABLE 1: ACS Closing Rules

1st Extremum	2nd Extremum	Closing Rule
XMAX	YMAX	Add line from (XMAX, Y of XMAX touch curve) to (XMAX, YMAX), then close with line from (XMAX, YMAX) to (X of YMAX touch curve, YMAX)
YMAX	XMIN	Add line from (X of YMAX touch curve, YMAX) to (XMIN, YMIN), then close with line from (XMIN, YMIN) to XMIN, Y of XMIN touch curve)
XMIN	YMIN	Add line from (XMIN, Y of XMIN touch curve) to (XMAX, YMAX) then close with line from (XMAX, YMAX) to (X of YMIN touch curve, YMAX)
YMIN	XMAX	Add line from (X of YMIN touch curve, YMAX) to (XMIN, YMAX) then close with line from (XMIN, YMAX) to (XMIN, Y of XMAX touch curve)
XMIN	XMIN	Add line from (XMIN, Y of 1st XMIN touch curve) to (XMIN, Y of 2nd XMIN touch curve)

1st Extremum	2nd Extremum	Closing Rule
XMAX	XMAX	Add line from (XMAX, Y of 1st XMAX touch curve) to (XMAX, Y of 2nd XMAX touch curve)
YMIN	YMIN	Add line from (X of 1st YMIN touch curve, YMIN) to (X of 2nd YMIN touch curve, YMIN)
YMAX	YMAX	Add line from (X of 1st YMAX touch curve, YMAX) to (X of 2nd YMAX touch curve)

With momentary reference to Figure 13L, note once again that the three boundary curves defining in part region 2092 intersect the main boundary at XMIN and YMAX. Thus, in this example, the system closes the curve set by adding horizontal and vertical lines in accordance with the second rule listed in Table 1 and as illustrated in Figure 13L.

Note that the solid atom which will ultimately be created through this subroutine will operate on the main solid as a Boolean difference, and will therefore be used to subtract either a corner or non-corner as specified in Step 1951.

Having effectively closed the curve set in the working view, the system then proceeds to Step 1954, wherein an axis of rotation is then defined according to whether the working view is the front or back view. In the front view, the axis of rotation is defined as the YMIN line. In the back view, the axis of rotation is defined as the YMAX line.

In visualizing the steps that follow, it is important to note that the key transformations will take place *out of* the drawing plane -- that is, much in the same way a finished section of framed wall is lifted from the ground and manipulated into place during a barn-raising, portions of the front or back views will be rotated out of the page and positioned advantageously with respect to the main solid. It should be remembered however, that the base solid was swept in such a way that it "exists" below the drawing plane.

Having defined the axis of rotation for the working view, the system then rotates the ACS 90 degrees clockwise about the YMIN axis for the front view or 90 degrees counter clockwise about the YMAX axis for the back view. Returning again to the example shown in Figure 13L, rotation of ACS 2092 about axis of rotation 2093 results in a closed curve 2094 lying in plane perpendicular to the drawing.

Next a 3x4 matrix is defined such that it operates to translate the ACS into place in front of the base solid (Steps 1956 and 1958). In essence, this step consists of two movements: (1) a translation of the ACS along the positive Y-axis by a distance equal to the difference between the active disjoint line YMIN and front view YMIN, and (2) a translation of the ACS along the negative Z-axis by a distance equal to the thickness of the base solid (determined earlier in Step 1016). Note that no X-axis translation is necessary because the front and back views have the same X range as the plan view. Finally, as shown in Step 1960, the ACS is swept along the length of the active disjoint line to form the solid atom which will ultimately be subtracted from the base solid (for example, atom 2097 in Figure 13M). The solid atom is added to the atom definitions structure as: ZTOP = YMIN of ADL, ZBOTTOM = YMAX of ADL, Work View number = 2 (Front) or 5(Back), BOOLE register = 2 (Difference). After thus defining the atom, the system exits the subroutine (Step 1962).

Referring now to Figure 13E, an exemplary method for processing vertical disjoint non-lines (Step 1914) will now be described. First, the system cycles through the topological constraint sets (TCSs) until the identifier for the active disjoint curve (ADC) is found (Steps 1970, 1972, and 1974). In a preferred embodiment, only tangent relations need be considered. If no ADC is found, the system sets ERROR register to "No Collinear Line" and returns (Step 1978). If the ADC has been found, the system proceeds to Step 1980 and extracts a tangent line (ATANLN) from the TCS. As shown in Step 1982, if ATANLN is in the top view, the system returns to Step 1970, and continues to cycle through the TCS. If ATANLN is in the top view, then the system proceeds to Step 1984, and finds the intersections of ADC with the top view main boundary. In Step 1986 the system defines the following parameters: ZDEL = length of ATANLN, ZTOP = 0.0, and ZBOTTOM = -ZDEL

The solid atom is then created by closing the ADC where it touches the top view main boundary and sweeping from ZTOP to ZBOTTOM. In this case, the work view will either be the front or back view, depending upon where ATANLN is found. After creating this atom, the system exits the subroutine (Step 1962).

Referring now to Figure 13F, an exemplary for processing horizontal disjoint lines (Step 1928) will now be described. As a preliminary matter, it should be clear that the method of processing horizontal disjoint lines will necessarily be analogous and in some

cases identical to the method used for vertical disjoint lines. For instance, while the front and back views were the most useful work views in the vertical context (from the standpoint of gathering sufficient information to generate an atom), the left end and right end views are the most useful in the horizontal context.

5 First, the system cycles through the first topological constraint set until the identifier for the active disjoint line (ADL) is found (Steps 2000, 2002, and 2004). If an ADL is found, the system proceeds to Step 2016. If no ADL has been found, and all TCSs have been processed, the system proceeds to Step 2008. In Step 2008, the system checks whether the ADL intersects with the intersection of two contiguous curves in either the
10 left end or right end views; if so, then an atom is created (Step 2008). Specifically, the system loops back to Step 1944 in Figure 13C and proceeds as before up to Step 1962.

Next, the system checks to see whether an atom was created (Step 2010), in which case the system exits the subroutine (Step 2014). If no atom was created, the ERROR register is set to "No Atom Created" (a non-fatal error), and the system exits.

15 If an ADL was found in Step 2004, the system finds another curve in the same TCS which is not in the current view (Step 2014). This view (either left end or right end) becomes the work view (Step 2018), and the matching collinear or tangent curve in the TCS is identified as CLTCURV (Step 2020).

Next, CLTCURV is examined (Steps 2022 and 2024). If CLTCURV is part of the
20 work view main boundary, the system proceeds to Step 2030. If CLTCURV is part of an interior closed curve set, the system proceeds to Step 2036, moves the closed curve set into the active curve set (ACS), then continues at Step 2038 (described below).

For work view main boundary curves (Step 2030) the system starts at CLTCURV and progresses forward and (if necessary) backward along the main boundary closed curve
25 set until the resulting chain of curves encounters exactly two from the set XMIN, YMIN, XMAX, and YMAX. Thus, the system identifies a short chain of elements whose outermost members terminate at the boundary of the main outline. The resulting boundary curves then are ordered such that the first Extremum touching curve is listed 1st, and the last Extremum touching curve is listed last within the active curve set.

30 In Steps 2032 and 2034, one or more "phantom curves" are added to the chain of boundary curves in order to close it off -- i.e., to create a closed curve set which will be

used to define the solid atom. Specifically, the active curve set (ACS) is closed using the rules set forth previously in Table 1.

Having thus defined the active curve set, the system then defines an axis of rotation (Step 2038). If the work view is the left end view, the vertical XMIN line is used; if the work view is the right end view, the vertical XMAX line is used.

Next, in accordance with step 2040, the ACS is rotated 90 degrees clockwise about the XMIN axis (for the left end view) or 90 degrees counter clockwise about the XMAX axis (for the right end view).

A 3x4 matrix is defined such that it operates to translate the ACS into place in front of the base solid (Steps 2042 and 2044). In essence, this step consists of two movements: (1) a translation of the ACS along the positive X-axis by a distance equal to the difference between the active disjoint line XMIN and the X value of all curves within the ACS, and (2) a translation of the ACS along the negative Z-axis by a distance equal to the thickness of the base solid (determined earlier in Step 1016). Note that no Y-axis translation is necessary because the left end and right end views have the same Y range as the plan view. Finally, as shown in Step 2046, the solid atom is defined as follows: ZTOP = YMIN of ADL, ZBOTTOM = YMAX of ADL, and BOOLE register = 2 (Difference)

Referring now to Figure 13I, an exemplary method for processing horizontal disjoint non-lines (Step 1930) will now be described. First, the system cycles through the topological constraint sets (TCSs) until the identifier for the active disjoint curve (ADC) is found (Steps 2050, 2052, and 2054). In a preferred embodiment, only tangent relations need be considered.

If no ADC is found, the system sets the ERROR register to "No Collinear Line" and returns (Step 2058). If the ADC has been found, the system proceeds to Step 2052 and extracts a tangent line (ATANLN) from the TCS. As shown in Step 2062, if ATANLN is in the top view, the system returns to Step 2050, and continues to cycle through the TCSs. If ATANLN is in the top view, then the system proceeds to Step 2064, and finds the intersections of ADC with the top view main boundary. In Step 2066 the system defines the following parameters: ZDEL = length of ATANLN, ZTOP = 0.0, and ZBOTTOM = -ZDEL.

The solid atom is then created by closing the ADC where it touches the top view main boundary and sweeping from ZTOP to ZBOTTOM. In this case, the work view will

either be the right end or left end views, depending upon where ATANLN is found. After creating this atom, the system exits the subroutine (Step 1962).

Referring now to Figure 13J, an exemplary method of processing oblique disjoint lines will now be described.

5 In Step 2072 the system determines the current view main boundary curve (CVMBC) that touches the oblique disjoint line (ODL). More specifically, the system addresses in turn each of the curves that comprise the main boundary of the current view in order to determine which curve or curves touch the ODL. In the case where more than one curve is a candidate for CVMBC, the system chooses the curve which is in the
10 direction of an extant oblique view. If more than one curve meets this additional criterion, then the choice of CVMBC is arbitrary, and the first one is suitably selected.

 Next, in Step 2073, the system cycles through the topological constraint sets until the identifier for the ODL is found. The first collinear, tangent, or 'endpoint on a curve' element related to ODLIN is identified as CLTCURV. It should be noted that while
15 more than one curve might satisfy these conditions, any one of the available curves may be chosen by the system for the purposes of generating the necessary solid atom. The system then checks whether CLTCURV is a member of an interior closed curve set. If not, then a nonfatal error is returned. If so, then the closed curve set is moved into the active curve set (ACS).

20 Having determined CVMBC, the system then defines a rotation point along an extension of ODLIN which falls outside the current view (Step 2074). The exact placement of this point is not critical, as long as it lies outside the current view. In a preferred embodiment, the point is defined to be 0.1 inches outside the current view.

 A rotation angle ROTANG is then computed as the angle that TMBCRV makes
25 with the positive X-Axis (Step 2076).

 Next, in Step 2078, the actual work view number corresponding to the appropriate oblique view is determined. Recall that available oblique views were numbered previously in Step 1010 starting at 7 and ending at $6+n$, where n is the total number of oblique views.

30 In Step 2084, a number of curves are rotated counterclockwise about the rotation point defined in Step 2074 by an angle ROTANG defined in Step 2076. This will effectively allow the system to process the oblique line exactly it processed vertical disjoint lines in Step 1912. More particularly, The following curves and curve sets are

rotated about the rotation point: 1) the top main boundary, 2) the entire oblique working view set, 3) the active curve set ACS, and 4) the oblique disjoint line ODL. This step effectively transforms the oblique view into a front view, referred to as the "pseudo-front view."

5 Having rotated and aligned the appropriate entities within the plane, an axis of rotation is defined in Step 2086 as the horizontal YMIN line in the pseudo-front view. The ACS is then rotated 90 degrees clockwise about this axis, then translated along the Y axis by a distance equal to the YMIN value of ODLIN minus the Y value of ACS within the pseudo-front view. Next, the ACS is translated along the negative Z axis by a distance
10 equal to the main solid thickness (determined earlier in Step 1016).

As shown in Step 2088, the parameters for the new atom are given by:
ZTOP=YMIN of ODLIN, and ZBOTTOM=YMAX of ODLIN.

Next, all rotated data plus all atom definition data (including ZTOP and ZBOTTOM) is rotated back clockwise about the rotation point by ROTANG degrees
15 (Step 2090). The work view number is then set to the oblique view number (Step 2092), and the system exits the subroutine (Step 2094).

Referring now to Figure 14A, an exemplary method for processing closed curve sets will now be described in detail. A count of closed curve sets is placed in the active register (Step 2102), and the first closed curve set identifier is extracted from the closed curve set
20 structure (structure 920 in Figure 9D). Next, in step 2110, the closed curve set XMAX curve and identifier are extracted from extrema structure (structure 1472 in Figure 8A).

The system then queries whether the XMAX curve is vertical (Step 2114). If so, then the system processes the vertical XMAX closed curve set (Step 2118). If XMAX is not vertical ("no" branch from Step 2114), the system proceeds to Step 2116, and
25 determines whether the XMAX curve is non-linear (e.g., a spline, conic, or arc). If the XMAX curve is found to be non-linear, then the system processes a non-linear XMAX closed curve set (Step 2120).

If at Step 2116 the system determined that the XMAX curve is not non-linear, the system then proceeds to Step 2126, and queries whether XMAX is a point on a curve. If
30 so, the system processes a point on a curve XMAX closed curve set (Step 2128). If in Step 2126 the system determined that XMAX is not a point on a curve, the system

extracts the closed curve set YMAX curve and identifier and places them in the active curve register and active curve identifier register respectively.

Next, the system successively queries whether YMAX is horizontal (Step 2136), non-linear (Step 2138), or a point on a curve (Step 2140). If YMAX is horizontal, the system processes a horizontal YMAX closed curve set (Step 2142). If YMAX is non-linear, the system processes a non-linear YMAX closed curve set (Step 2144). If YMAX is a point on a curve, the system processes a point-on-a-curve YMAX closed curve set (Step 2146).

The system flow provides that, after each of the major processing subroutines have been completed (i.e., Steps 2118, 2120, 2128, 2136, 2138, and 2140), the system interrogates whether the ERROR register is zero. If not, then the system proceeds to Step 2108, described below. If the ERROR register was found to be zero in Steps 2122, 2124, 2148, 2150, or 2152, then the system proceeds to Step 2104. In this step the system adds the resulting axial sweep atom to the atom definitions structure then proceeds to Step 2108.

In Step 2108, the system determines whether all closed curve sets have been processed. If so, the system exits this subroutine (Step 2112); if not, then the system loops back to Step 2106, extracts the next closed curve identifier from the closed curve set structure, and proceeds as before until all closed curve sets have been processed.

Referring now to Figure 14C, the method of processing vertical XMAX closed curve sets will now be described in detail (Step 2118). First, the system cycles through the first collinear topological constraint set TCS until the identifier for the active XMAX line (AXL) is found (Step 2154). If a collinear AXL has been found, the system continues to Step 2164. If no collinear AXL has been found after processing all TCSs (Step 2158), then the system indicates that no atom was created (a non-fatal error), and exits the subroutine (Step 2162).

If a collinear AXL was previously found ("yes" branch from Step 2156), then the system finds another curve in the same TCS which is not in the current view (Step 2164). Referring now to Figure 14D, the system then decides which view will be designated the work view (Step 2168). More particularly, the system designates the non-top view as the work view for the new atom if and only if: 1) the top view closed curve set is a rectangle parallel to the X-axis, and 2) the front (or back) view line collinear with AXL is a part of

a closed curve set which is not a rectangle parallel to the X-axis. If one or both of these criteria are not met, then the top view becomes the work view and the system proceeds to Step 2174. If both criteria are met, the system proceeds to Step 2172 and moves the closed curve set into the active curve set (ACS) and the top view XMAX line identifier into the active disjoint line register (Step 2172). That is, in this special case (when both criteria are satisfied) we can effectively treat the XMAX line as an active disjoint curve and process it in accordance with the procedure described above in conjunction with Figure 13H. Specifically, the system proceeds back to Step 2038 in Figure 13H and processes the line accordingly.

If both criteria set forth in Step 2168 were not satisfied, the system branches to Step 2174 via Step 2170. In Step 2174, the system defines the top view as the work view and designates the line collinear with AXL as "COLINLN" (the length of which is ZDEL). Next, the system queries whether the top view closed curve set is "inside" a larger closed curve set. If not, then the system establishes 0.0 as the "local top." That is, such a closed curve set effectively resets the local top extrema for the purposes of atom generation. Next, the system determines whether COLINLN YMAX is above the local top. If so then the resulting solid atom is defined as: BOOLE register = 1, ZBOTTOM = local top, and ZTOP = ZBOTTOM + ZDEL. The system then sets "local top" as ZTOP, then exits the subroutine (Step 2186).

If at Step 2180 COLINLN YMAX was found to be above the local top, then the system defines the solid atom as: BOOLE register = 2, ZTOP = local top, and ZBOTTOM = ZTOP - ZDEL. After defining the solid atom, the system sets local top = ZBOTTOM then exits the subroutine.

Referring now to Figure 14E, an exemplary method of processing horizontal YMAX closed curve sets will now be described in detail (Step 2142). First, the system cycles through the first collinear topological constraint set until the identifier for the active YMAX line (AYL) is found (Step 2188). If a collinear AYL has been found, the system continues to Step 2198. If no collinear AXL has been found after processing all TCSs (Step 2192), then the system indicates that no atom was created (a non-fatal error), and exits the subroutine (Step 2196).

If a collinear AXL was previously found ("yes" branch from Step 2190), then the system finds another curve in the same TCS which is not in the plan view (Step 2198).

Referring now to Figure 14F, the system then decides which view will be designated the work view (Step 2202). More particularly, the system designates the non-top view as the work view for the new atom if and only if: 1) the top view closed curve set is a rectangle parallel to the X-axis, and 2) the left (or right) view line collinear with AYL is a part of a closed curve set which is not a rectangle parallel to the X-axis. If one or both of these criteria are not met, then the top view becomes the work view and the system proceeds to Step 2204. If both criteria are met, the system proceeds to Step 2206 and moves the closed curve set into the active curve set (ACS) and the top view YMAX line identifier into the active disjoint line register (Step 2206). That is, in this special case (where both criteria are satisfied) we can effectively treat the YMAX line as an active disjoint curve and process it in accordance with the procedure described above. Thus, the system proceeds to Step 2030 in Figure 13G and continues to Step 2048, where the subroutine exits.

If both criteria set forth in Step 2202 are not satisfied, then the system branches to Step 2208 via Step 2204. In Step 2208, the system defines the top view as the work view and designates the line collinear with AYL as "COLINLN" (the length of which is ZDEL). Next, the system queries whether the top view closed curve set is "inside" a larger closed curve set. If not, then the system establishes 0.0 as the "local top." Next, the system determines whether COLINLN XMAX is above the local top. If so then the resulting solid atom is defined in Step 2214 as: BOOLE register = 1, ZBOTTOM = local top, and ZTOP = ZBOTTOM + ZDEL. The system then sets "local top" as ZTOP, then exits the subroutine (Step 2220).

If at Step 2214 COLINLN YMAX was found not to be above the local top, then the system in Step 2218 defines the solid atom as: BOOLE register = 2, ZTOP = local top, ZBOTTOM = ZTOP - ZDEL. After defining the solid atom, the system sets local top = ZBOTTOM then exits the subroutine.

Referring now to Figure 14G, an exemplary method for processing non-linear XMAX closed curve sets will now be described in detail (Step 2120). First, the system cycles through the first tangent TCS until the identifier for the active XMAX curve (AXC) is found (Step 2222). If, after examining all TCSs, the system fails to find a tangent AXC, then the system indicates that no atom was created (non-fatal error), end exits the subroutine (Steps 2226, 2228, and 2230).

If previously at Step 2224 the system successfully found a tangent AXC, then the system proceeds to Step 2232 and locates another curve in the same TCS which is not in the current view. As will be seen in Steps 2234 through 2248, this non-top view curve will be used to provide the ZTOP, ZBOTTOM, and length of the new solid atom.

5 Referring now to Figure 14H, the top view is designated as the work view, and the line tangent to AXC is labeled TANLINE. ZDEL is set equal to the length of TANLINE. Next, the system queries whether the top view closed curve set is "inside" a larger closed curve set (Step 2238). If not, the system proceeds to Step 2240 and establishes 0.0 as the local top. From Step 2240, the system proceeds to Step 2242, described in detail below.

10 If the top view closed curve set was found to be inside a larger closed curve set ("yes" branch from Step 2238), the system determines whether the TANLINE YMAX is above the local top (Step 2242). If so, the solid atom is defined in Step 2244 as: BOOLE register = 1, ZBOTTOM = local top, ZTOP = ZBOTTOM+ZDEL. The local top is then set to ZTOP, and the system exits the subroutine (Step 2248).

15 When the system determines that the TANLINE YMAX is not above the local top ("no" branch from Step 2242), the solid atom is defined in Step 2246 as: BOOLE register = 2, ZBOTTOM = ZTOP-ZDEL, and ZTOP = Local top. The local top is then set to ZBOTTOM, and the system exits the subroutine (Step 2248).

Referring now to Figure 14I, an exemplary method for processing point-on-a-curve XMAX closed curve sets will now be described in detail (Step 2128). First, the system cycles through the first point-on-a-curve TCS until the identifier for the active XMAX curve (AXC) is found (Step 2250). If, after examining all TCSs, the system fails to find a point-on-a-curve AXC, then the system indicates that no atom was created (non-fatal error), end exits the subroutine (Steps 2254, 2256, and 2258).

25 If previously at Step 2252 the system successfully found a point-on-a-curve AXC, then the system proceeds to Step 2260 and locates another curve in the same TCS which is not in the plan view. As will be seen in Steps 2262 through 2276, this non-top view curve will be used to provide the ZTOP, ZBOTTOM, and length of the new solid atom.

Referring now to Figure 14J, the top view is designated as the work view, and the line related to AXC as an end-point-on-a-curve is labeled ENDPTLINE. ZDEL is set equal to the length of ENDPTLINE. Next, the system queries whether the top view closed curve set is "inside" a larger closed curve set (Step 2266). If not, the system proceeds to

Step 2268 and establishes 0.0 as the local top. From Step 2268, the system proceeds to Step 2270, described in detail below.

If the top view closed curve set was found to be inside a larger closed curve set ("yes" branch from Step 2266), the system determines whether the ENDPTLINE YMAX is above the local top (Step 2270). If so, the solid atom is defined in Step 2272 as: BOOLE register = 1, ZBOTTOM = local top, and ZTOP = ZBOTTOM+ZDEL. The local top is then set to ZTOP, and the system exits the subroutine (Step 2248). When the system determines that the ENDPTLINE YMAX is not above the local top ("no" branch from Step 2270), the solid atom is defined in Step 2274 as: BOOLE register = 2, ZBOTTOM = ZTOP-ZDEL, and ZTOP = Local top. The local top is then set to ZBOTTOM, and the system exits the subroutine (Step 2276).

Referring now to Figure 15A, an exemplary method for processing open curve sets will now be described in detail (Step 1026 in Figure 4). Initially, in Step 2302, the count of open curve sets is placed into the active register. The system then extracts the identifier of the first open curve set from open curve set structure. This becomes the active open curve set (AOPNCS). Next, the system finds the curve in the plan view main profile which the AOPNCS touches (referred to as the "touching" curve), then determines its orientation and whether the resulting atom should be added or subtracted from the base solid (Step 2314). This step is discussed in further detail below in conjunction with Figure 15M.

Having determined the orientation and appropriate Boolean operation of the "touching" curve, the system then queries whether the ACPNCS is semi-rectangular. As used here, semi-rectangular means either 1) an open curve set comprising two line segments defining a right angle, or 2) an open curve set comprising three line segments which define three sides of a rectangle (i.e., opposite sides which are parallel, of equal length, and perpendicular to the third side).

If AOPNCS is not semi-rectangular ("no" branch from Step 2316), the system processes the open curve set which using the 2nd view main boundary (Step 2318). This process is described in further detail below in conjunction with Figures 15Q and 15P. If AOPNCS is semi-rectangular, the system proceeds to Step 2322 and queries whether the "touching" curve is horizontal. If so, the system processes a horizontal touching curve (Step 2324); if not, then the system proceeds to Step 2326 and determines whether the

"touching" curve is vertical. If the "touching" curve is vertical, the system proceeds to Step 2328 and processes a vertical touching curve. If it is not vertical, the system processes an oblique touching curve (Step 2330).

5 After each of Steps 2324, 2328, and 2330 have been completed -- corresponding to the processing of horizontal, vertical, and oblique curves respectively -- the system advances to Step 2308 and queries whether all open curve sets have been processed. If so, the system exits the subroutine (Step 2312); if not, the system proceeds to Step 2306, extracts the next open curve set identifier, and continues as above until the list of open curve sets has been exhausted.

10 Having described the main loop associated with an exemplary method of processing open curve sets, the individual subroutines used to process the various classes of open curve sets will now be described in detail.

An exemplary method of processing open curve sets satisfying the horizontal "touching" curve condition will now be described in conjunction with Figure 15C. First, 15 in Step 2332, the system cycles through the TCS until the identifier for the open curve set XMAX (OCSXMAX) is found. If the OCSXMAX has been found, the system branches from Step 2334 and proceeds to Step 2342. If the OCSXMAX has not been found after all TCSs have been processed (Step 2336), the system sets the ERROR register to "NO CONSTRAINT" found (Step 2338), and exits the subroutine.

20 Referring now to Figure 15D, in the case where OCSXMAX has been found ("yes" branch in Step 2334), the system finds another curve in the same TCS which is not in the plan view (Step 2344). The work view is set to the non-plan view curve found in the TCS. Note that since OCSXMAX is a vertical line, the work view must be front or back.

25 In Step 2346, the type of constraint is determined from the TCS, and the curve matching OCSXMAX is identified as MATCHCURV.

Next, the system determines whether MATCHCURV is an element of an open curve set (Step 2348). If not, the system moves the closed curve set into the active curve set (ACS), then advances to Step 2370, described below.

30 In the case where MATCHCURV was found to be an element in an open curve set ("yes" branch in Step 2348), the system proceeds to Step 2352 and queries whether MATCHCURV touches the work view boundary in two places. If not, the ERROR register is set to "Bad Open Curve Set," and the system exits the subroutine (Step 2354).

It should be noted that it is topologically possible to have within a drawing a MATCHCURV which is part of an open curve set but which does not touch the work view boundary in two places. Nevertheless, this situation -- while clearly comprehended by the present invention -- is deemed by the present inventor to be relatively uncommon, and hence it is not included in the illustrated exemplary embodiment.

Referring now to Figure 15E, when MATCHCURV is found to touch the work view boundary in two places ("yes" branch in Step 2352), the system queries whether the work view open curve set is a semi-rectangle (Step 2358). If not, the system proceeds to Step 2356, where the work view open curve set is closed with a line extending from the first to second end. This closed curve is then moved into the ACS and the system proceeds to Step 2370, described below.

If, in Step 2358, the work view open curve set is found to be a semi-rectangle, then the system proceeds to Step 2362 and defines the atom to be created. More particularly, the system defines an atom where: $Z_{TOP} = 0$, and the length of sweep = length of MATCHCURV. This atom is then added as an axial sweep to the atom definition structure (Step 2364) and the system exits the subroutine (Step 2366).

After closing the work view open curve set in accordance with Step 2356, or moving the closed curve set into the ACS in accordance with Step 2360, the system proceeds to Step 2370 and employs a rotation of ACS out of the drawing plane in order to position the curve set advantageously with respect to the base solid. More particularly, the system rotates the ACS 90 degrees clockwise about the YMIN axis for the front view, or 90 degrees counter clockwise about the YMAX axis for the back view.

Next, in Step 2368, a 3x4 unit matrix is defined to produce a spatial translation of the ACS, wherein: $\Delta X = 0.0$, $\Delta Y = OCSX_{MAX} Y_{MIN} - YVAL$, and $\Delta Z = -SOLTHICK$. This operation is applied to the ACS (Step 2372), then the resulting solid atom is defined as: $Z_{TOP} = OCSX_{MAX} Y_{MIN}$, $Z_{BOTTOM} = OCSX_{MAX} Y_{MAX}$, and Work View register = 2 (Front) or 5 (Back). After defining the solid atom, the system exits the subroutine (Step 2376).

An exemplary method of processing open curve sets satisfying the vertical "touching" curve condition will now be described in conjunction with Figure 15F. First, in Step 2378, the system cycles through the TCS until the identifier for the open curve set YMIN (OCSYMIN) is found. If the OCSYMIN has been found, the system branches

from Step 2384 and proceeds to Step 2392. If the OCSYMIN has not been found after all TCSs have been processed (Step 2382), the system sets the ERROR register to "NO CONSTRAINT FOUND" (Step 2384), and exits the subroutine (Step 2386).

5 Referring now to Figure 15G, in the case where OCSYMIN has been found ("yes" branch in Step 2380), the system finds another curve in the same TCS which is not in the plan view (Step 2392). The work view is set to the non-plan view curve found in the TCS. Note that since OCSYMIN is a vertical line, the work view must be left or right.

In Step 2396, the type of constraint is determined from the TCS, and the curve matching OCSYMIN is identified as MATCHCURV.

10 Next, the system determines whether MATCHCURV is an element in an open curve set (Step 2398). If not, the system moves the closed curve set into the active curve set (ACS), then advances to Step 2420, described below.

15 In the case where MATCHCURV was found to be an element in an open curve set ("yes" branch in Step 2398), the system proceeds to Step 2402 and queries whether MATCHCURV touches the work view boundary in two places. If not, the ERROR register is set to "Bad Open Curve Set," and the system exits the subroutine (Steps 2400 and 2404).

20 Referring now to Figure 15H, when MATCHCURV is found to touch the work view boundary in two places ("yes" branch in Step 2402), the system queries whether the work view open curve set is a semi-rectangle (Step 2408). If not, the system proceeds to Step 2406, where the work view open curve set is closed with a line extending from the first to second end. This closed curve is then moved into the ACS and the system proceeds to Step 2420, described below.

25 If, in Step 2408, the work view open curve set was found to be a semi-rectangle, then the system proceeds to Step 2412 and defines the atom to be created. More particularly, the system creates an atom where: $Z_{TOP} = 0$, Length of sweep = length of MATCHCURV. This atom is then added as an axial sweep to the atom definition structure (Step 2414) and the system exists the subroutine (Step 2416).

30 After closing the work view open curve set in accordance with Step 2406, or moving the closed curve set into the ACS in accordance with Step 2410, the system proceeds to Step 2420 and employs a rotation of ACS out of the drawing plane in order to position the curve set advantageously with respect to the base solid. More particularly,

the system rotates the ACS 90 degrees clockwise about the XMIN axis for the left view, or 90 degrees counter clockwise about the XMAX axis for the right view.

Next, in Step 2418, a 3x4 unit matrix is defined to produce a spatial translation of the ACS, wherein: $\Delta X = XVAL - OCSYMIN \cdot XMAX$, $\Delta Y = 0.0$, AND $\Delta Z = -SOLTHICK$. This operation is applied to the ACS (Step 2422), then the resulting solid atom is defined as: $ZTOP = OCSXMIN \cdot XMIN$, $ZBOTTOM = OCSYMIN \cdot XMAX$, and Work View register = 4 (Left) or 3 (Right). After defining the solid atom, the system exits the subroutine (Step 2426).

An exemplary method of processing open curve sets satisfying the oblique "touching" curve condition (referred to now as TOUCHING) will now be described in conjunction with Figure 15I. First, in Step 2428, the system cycles through the TCS until the identifier for the open curve set end (OCSEND) is found. If the OCSEND has been found, the system branches from Step 2430 and proceeds to Step 2438. If the OCSEND has not been found after all TCSs have been processed (Step 2432), the system sets the ERROR register to "NO CONSTRAINT" found (Step 2434), and exits the subroutine (Step 2436).

Referring now to Figure 15J, in the case where OCSEND has been found ("yes" branch in Step 2430), the system finds another curve in the same TCS which is not in the plan view (Step 2438). The work view is set to the non-plan view curve found in the TCS. Note that since OCSEND is an oblique line, the work view must be an oblique view.

In step 2442, the type of constraint is determined from the TCS, and the curve matching OCSEND is identified as MATCHCURV.

Next, the system determines whether MATCHCURV is an element in an open curve set (Step 2444). If not, the system moves the closed curve set into the active curve set (ACS), then advances to Step 2464, described below.

In the case where MATCHCURV was found to be an element in an open curve set ("yes" branch in Step 2444), the system proceeds to Step 2448 and queries whether MATCHCURV touches the work view boundary in two places. If not, the ERROR register is set to "Bad Open Curve Set," and the system exits the subroutine (Steps 2446 and 2450).

Referring now to Figure 15K, when MATCHCURV is found to touch the work view boundary in two places ("yes" branch in Step 2448), the system queries whether the

work view open curve set is a semi-rectangle (Step 2454). If not, the system proceeds to Step 2452, where the work view open curve set is closed with a line extending from the first to second end. This closed curve is then moved into the ACS and the system proceeds to Step 2464, described below.

5 If, in Step 2454, the work view open curve set was found to be a semi-rectangle, then the system proceeds to Step 2458 and defines the atom to be created. More particularly, the system creates an atom where: $ZTOP = 0$, and length of sweep = length of MATCHCURV. This atom is then added as an axial sweep to the atom definition structure (Step 2460) and the system exists the subroutine (Step 2462).

10 After closing the work view open curve set in accordance with Step 2452, or moving the closed curve set into the ACS in accordance with Step 2456, the system proceeds to Step 2464.

 In Step 2464 the system defines a rotation point along an extension of OCSEND which falls outside the current view. The exact placement of this point is not critical, as long as it lies outside the current view. In a preferred embodiment, the point is defined to be 0.1 inches outside the current view.

15 After defining a rotation point, a rotation angle is defined as the angle that MATCHCURV makes with the positive X-axis (Step 2466). The system then determines the actual work view number (i.e., the number corresponding to the particular oblique view of interest) then rotates a number of curves and curve sets counter clockwise about the rotation point by ROTANG degrees. The curves rotated about this point suitably comprise: 1) the top main boundary, 2) the entire oblique view, 3) the active curve set, and 4) the OCSEND curve.

20 Having rotated the oblique view such that it has effectively become the front view, or "pseudo-front view," the system then defines an axis of rotation as the horizontal YMIN line in the pseudo front view (Step 2472). Next, the system rotates the ACS 90 degrees clockwise about the axis of rotation. A 3x4 unit matrix is defined to produce a spatial translation of the ACS, wherein: $\Delta X = 0.0$, $\Delta Y = (OCSEND \text{ pseudo } YMIN) - YVAL$, and $\Delta Z = -SOLTHICK$. This translation operation is applied to the ACS, allowing us to define the solid atom as: $ZTOP = OCSEND \text{ } YMIN$, and $ZBOTTOM = OCSYMIN \text{ } YMAX$.

After defining the solid atom, the system rotates all previously rotated data (including ZTOP and ZBOTTOM) clockwise about the rotation point ROTANG degrees (Step 2476). The system sets the work view register to the oblique view number (Step 2478), then exits the subroutine (Step 2478).

5 Referring now to Figure 15M, an exemplary method for determining the TOUCHING curve orientation and whether the atom should be added or subtracted from the base solid will now be described in detail (this step corresponds to Step 2314 in Figure 15A).

10 In Step 2504 the system determines whether the first active open curve set (AOPNCS) curve touches the plan view boundary. If so, the system designates the first AOPNCS curve as the TOUCHING curve; if not, the then last AOPNCS curve is designated as such.

15 Next, in Step 2506, the orientation of the TOUCHED curve is determined (i.e., either vertical, horizontal, or oblique). The TOUCHED curve is simply that curve which the TOUCHING curve makes contact with along the plan view main boundary.

20 In the steps that follow, the system determines whether the atom to be created is inside or outside the base solid. As a preliminary matter, a number of registers are defined and initialized in Step 2508. Specifically, the curve identifier for the TOUCHED current view curve is placed in the TESTCRV register, the curve following (i.e. next in the list after) TESTCRV in the current view boundary is placed in the NEXTCRV register, the TESTCRV endpoints are placed into the TESTENDS register, and the NEXTCRV endpoints are placed in the NEXTENDS register.

25 Having initializing the registers, the system then proceeds to Step 2510 (Figure 15N), where a branching occurs according to the orientation of the TOUCHED curve. More particularly, when the TOUCHED curve is horizontal, the system proceeds to Step 2534; when the TOUCHED curve is vertical, the system proceeds to Step 2512; and when the TOUCHED curve is oblique, the system proceeds to Step 2534.

Referring now to Figure 15N, an exemplary method for examining horizontal TOUCHED curves (Step 2512) will now be described in detail.

30 First, the system queries NEXTCRV and determines whether the X-value of its endpoints are greater than TESTCRV (Step 2516). If so, then the system concludes that the "inside" of the closed curve is in the positive Y direction (Step 2514); if not, the

system concludes that the inside of the closed curve is in the negative Y direction (Step 2520).

5 When the inside is determined to be in the positive Y direction ("yes" branch in Step 2516), the system continues to Step 2518 and queries whether TOUCHING contacts the main boundary at its YMIN end. If so, then a difference operation for the inside region is called for, and the system sets the BOOLE register to 2 (Step 2522); if not, then a union operation for the outside region is called for, and the system sets the BOOLE register to 1 (Step 2530). After setting the value of the BOOLE register in either of Steps 2522 or 2530, the system exits the subroutine (Step 2526).

10 When the inside is determined to be in the negative Y direction ("no" branch in Step 2516), the system continues to Step 2524 and queries whether TOUCHING contacts the main boundary at its YMIN end. If so, then a union operation for the outside region is called for, and the system sets the BOOLE register to 1 (Step 2528); if not, then a difference operation for the inside region is called for, and the system sets the BOOLE register to 2 (Step 2532). After setting the value of the BOOLE register in either of Steps 15 2528 or 2532, the system exits the subroutine (Step 2526).

Referring now to Figure 150, an exemplary method for examining vertical TOUCHED curves (Step 2534) will now be described in detail.

20 First, the system queries NEXTCRV and determines whether the Y-value of its endpoints are greater than TESTCRV (Step 2538). If so, then the system concludes that the "inside" of the closed curve is in the negative X direction (Step 2536); if not, the system concludes that the outside of the closed curve is in the positive X direction (Step 2542).

25 When the inside is determined to be in the negative X direction ("yes" branch in Step 2538), the system continues to Step 2540 and queries whether TOUCHING contacts the main boundary at its XMIN end. If so, then a union operation for the outside region is called for, and the system sets the BOOLE register to 1 (Step 2544); if not, then a difference operation for the inside region is called for, and the system sets the BOOLE register to 2 (Step 2552). After setting the value of the BOOLE register in either of Steps 30 2552 or 2544, the system exits the subroutine (Step 2548).

When the outside is determined to be in the positive X direction ("no" branch in Step 2538), the system continues to Step 2546 and queries whether TOUCHING contacts

the main boundary at its XMIN end. If so, then a difference operation for the inside region is called for, and the system sets the BOOLE register to 2 (Step 2550); if not, then a union operation for the outside region is called for, and the system sets the BOOLE register to 1 (Step 2554). After setting the value of the BOOLE register in either of Steps 2554 or
5 2550, the system exits the subroutine (Step 2548).

Referring now to Figure 15P, an exemplary method for examining oblique TOUCHED curves (Step 2556) will now be described in detail.

In Step 2560, the system first determines whether NEXTCRV starts with a larger Y value. If so, the system proceeds to Step 2562; if not, the system proceeds to Step
10 2558.

When a larger Y value does not start NEXTCRV ("no" branch in Step 2560), the system determines whether the slope of the oblique curve is positive (Step 2558). If so, the system concludes that the inside of the closed curve corresponds to the positive X and positive Y directions (Step 2566); if not, the system concludes that the inside of the closed
15 curve corresponds to the negative X and positive Y directions (Step 2564). After thus determining the direction of the inside region, the system proceeds to Step 2572 and queries whether the TOUCHING curve contacts the main boundary at its XMIN end. If so, the system concludes that a union operation is appropriate for the outside region, and sets the BOOLE register to 1 (Step 2576); if not, the system uses a difference operation
20 for the inside region and sets the BOOLE register to 2. After thus setting the BOOLE register in Steps 2576 and 2584, the system exits the subroutine (Step 2580).

When a larger Y value starts NEXTCRV ("yes" branch in Step 2560), the system determines whether the slope of the oblique curve is positive (Step 2562). If so, the system concludes that the inside of the closed curve corresponds to the minus X and
25 minus Y directions (Step 2568); if not, the system concludes that the inside of the closed curve corresponds to the negative X and positive Y directions (Step 2570). After thus determining the direction of the inside region, the system proceeds to Step 2574 and queries whether the TOUCHING curve contacts the main boundary at its XMIN end. If so, the system concludes that a union operation is appropriate for the outside region, and
30 sets the BOOLE register to 1 (Step 2578); if not, the system uses a difference operation for the inside region and sets the BOOLE register to 2. After thus setting the BOOLE register in Steps 2578 and 2586, the system exits the subroutine (Step 2580).

Referring now to Figures 15Q and 15R, an exemplary method of processing open curve sets which uses the "2nd view" main boundary (Step 2318) will now be described. First, it will be appreciated that since the open set is not semi-rectangular, the 2nd view can only provide the ZTOP and ZBOTTOM for a top view sweep. Thus, in Step 2610, the system queries whether the open set is mostly horizontal; that is, whether the curve set, taken as a whole, has greater X-axis extrema than Y-axis extrema (or, more succinctly, whether it's wider than it is tall). If the open set is not mostly horizontal, the system proceeds to Step 2608, discussed below. Otherwise, the system establishes the longest horizontal line (HORIZLINE) in the open curve set and identifies the top view boundary curve it touches (Step 2616). This curve is thereafter referred to as TOUCHING. Next, in Step 2622, the system cycles through the first TCS until the identifier for HORIZLINE is found. If HORIZLINE is not found, the system loops back to Step 2622 via Step 2624 until all TCSs are processed. In the event HORIZLINE is still not found, the system sets the ERROR register to "NO CONSTRAINT" (Step 2618), then exits the subroutine (Step 2612). If previously at Step 2628 the system successfully located HORIZLINE in a TCS, the system proceeds to Step 2630, discussed below.

If the open set was not found to be mostly horizontal ("no" branch in Step 2610), the system queries whether the open set is mostly vertical (Step 2608j). If not, the system indicates that no atom was defined (Step 2604) and returns (Step 2602). If the open set is found to be mostly vertical, then the system establishes the longest vertical line (VERTLINE) in the open curve set and identifies the top view boundary curve that it touches (Step 2614). This curve is identified as TOUCHING. In Step 2620, then, the system cycles through the first TCS until the identifier for VERTLINE is found. If VERTLINE is not found, the system cycles through the remaining TCSs until all have been processed (Step 2624). If VERTLINE has still not been found after examining all TCSs, the system sets the ERROR register to "NO CONSTRAINT" and exits (Steps 2618 and 2612).

In the case where either VERTLINE or HORIZLINE has been found ("yes" branch in Step 2626 and "yes" branch in Step 2628), the system in Step 2630 then focuses on a subset of constraints, suitably consisting of colinear, tangent, and endpoint-on-a-curve. The curve matching HORIZLINE or VERTLINE in the work view is identified as MATCHCURV.

Next, the system queries whether MATCHCURV is an element in the work view main boundary (Step 2632). If not, the system sets the ERROR register to "No atom defined" and returns (Steps 2634 and 2638).

Where MATCHCURV is an element in the work view main boundary, the system searches forward (and backward if necessary), starting with MATCHCURV, until a combination of exactly two from the set XMIN, YMIN, XMAX, and YMAX have been encountered (Step 2636). The system then closes the open curve set (Step 2640). This closing is done according to the nature of the extrema encountered in Step 2636. More particularly, if the first extremum is a YMAX or YMIN, and the second extremum is an XMIN, then the system adds a line from the (X, Y) point where TOUCHING and VERTLINE contact to a point defined as (Top XMIN, Y where TOUCHING and VERTLINE contact). Another line is added from the previous point to a point defined as (Top XMIN, opposite end of OCS Y from start). A final line is then added from the previous point to a point defined by the (X,Y) coordinates of the opposite end of OCS from the start point.

If the first extremum is a YMAX or YMIN, and the second extremum is an XMAX, then the system adds a line from the (X, Y) point where TOUCHING and VERTLINE contact to a point defined as (Top XMIN, Y where TOUCHING and VERTLINE contact). Another line is added from the previous point to a point defined as (Top XMIN, opposite end of OCS Y from start). A final line is then added from the previous point to a point defined by the (X,Y) coordinates of the opposite end of OCS from the start point.

It will be appreciated that since MATCHCURV is a line, both extremum cannot be the same. It will also be appreciated that although these closing rules are set forth above based on the front view, the same rules may be employed in other views through a trivial mapping.

After thus closing the open curve set, the system defines a solid atom as: BOOLE = 1, ZDEL = length of VERTLINE (or HORIZLINE), ZTOP = 0.0, and ZBOTTOM = -ZDEL (Step 2642). At this point, the system returns to previously described Step 2304 (Figure 15A).

At this point, all disjoint, closed, and open curve sets have been processed and the respective solid atoms have been cataloged in atom definitions structure 3000. The system then, in Step 1032, takes the first two atoms (atom number one being the base solid

generated previously in Step 1018) and, in accordance with methods well known in the art, applies the corresponding Boolean operation to the base solid. The resulting solid then becomes the new working base solid, and the system continues through the list of atoms until the finished geometric solid is created.

5 Although the present invention is set forth herein in the context of the appended drawing figures, it should be appreciated that the invention is not limited to the specific forms shown. Various other modifications, variations, and enhancements in the design and arrangement of the systems and methods discussed herein may be made in the context of the present invention. For example, a preferred embodiment of the present invention is
10 described herein in the context of a traditional computer system operating in a sequential mode; however, the present invention may be advantageously utilized in a parallel processing environment.

 In addition, while an exemplary embodiment of the present invention was described in the context of generating a single contiguous solid, it will be appreciated that the
15 described methods may be utilized for generating multiple solids or assemblies. More particularly, two-dimensional drawing data portraying multiple parts -- for example, exploded views of interacting components -- may be converted into a group of corresponding geometric solids.

 These and other modifications may be made in the design and implementation of
20 various aspects of the invention without departing from the spirit and scope of the invention as set forth in the appended claims.

Claims

- 1 1. A method of generating and displaying a three-dimensional geometric solid
2 from two-dimensional drawing data using a digital computer, wherein said two-
3 dimensional drawing data comprises a plurality of geometric entities, comprising the steps
4 of:
5 segregating said geometric entities into a plurality of respective view sets;
6
7 automatically identifying a plurality of relational constraints associated with
8 said geometric entities; and
9 automatically generating and displaying said three-dimensional geometric
10 solid in accordance with said segregating and defining steps.
- 1 2. The method of claim 1, wherein said segregating step comprises the steps
2 of:
3 grouping said geometric entities into closed curve sets, open curve sets and
4 disjoint curves;
5 ordering said closed curve sets as a function of area;
6 computing a center of gravity for each of said closed curve sets;
7 grouping at least one of said closed curve sets into a unique one of said
8 view sets in accordance with said ordering step, said computing step, and
9 whether said curve set makes contact with another of said closed curve sets;
10 grouping at least one of said open curve sets into a unique one of said view
11 sets by identifying a closed curve set with which it makes contact;
12 and
13 grouping each of said disjoint curves into a unique one of said view sets by
14 identifying a closed curve set with which it makes contact.

3. The method of claim 1, wherein said two-dimensional drawing data comprises implicit view set information, and said segregating step comprises the step of using said implicit view set information to group said geometric entities into said plurality of view sets.

4. The method of claim 1, wherein said segregating step comprises the step of prompting, using a graphical user interface associated with said digital computer, a user to select sets of geometric entities.

1 5. The method of claim 1, wherein said identifying step comprises the steps of:
2 compiling, for each of said view sets, a table of relational constraints
3 associated with geometric entities within said view set; and
4 compiling, for each of said view sets, a table of relational constraints
5 associated with geometric entities in other view sets.

1 6. The method claim of claim 1, wherein each of said relational constraints
2 comprise:
3 parallel;
4 normal;
5 collinear lines;
6 tangent;
7 horizontal;
8 vertical;
9 symmetrical;
10 collinear endpoints;
11 endpoints on a curve;
12 concentric arcs;
13 coincident arcs;
14 fixed;
15 point on two curves;
16 coincident points;
17 equal radius arcs;
18 equal length lines; and
19 equal distance.

1 7. The method of claim 1, wherein said generating step comprises the steps of:
2 ordering said view sets as a plan view set and at least one second view set;

3 grouping said geometric entities within said plan view set and said second
4 view set into closed curve sets, open curve sets and disjoint curves;
5 determining a base solid type;
6 defining a base solid in accordance with said base solid type;
7 defining a plurality of solid atoms; and
8 creating said three-dimensional geometric solid in accordance with said base
9 solid and said plurality of solid atoms.

8. The method of claim 7, wherein said ordering step comprises the step of
prompting a user to manually order said view sets using a graphical user interface.

1 9. The method of claim 7, wherein said ordering step comprises the steps of:
2 computing a center of gravity for each of said view
3 sets;
4 identifying a plan view set in accordance with user input;
5 and
6 ordering said second view set in accordance with the
7 position of said plan view set center of gravity with
8 respect to the position of said second view set center
9 of gravity.

1 10. The method of claim 7, wherein said grouping step comprises the steps of:
2 computing endpoints for at least one of said geometric entities;
3 defining at least one first subset of said geometric entities as a closed curve
4 set when said first subset forms a closed chain of curves having no
5 open endpoints;
6 defining at least one second subset of said geometric entities as an open
7 curve set when said second subset forms a chain of curves having a
8 first open endpoint and a second open endpoint; and
9 defining at least one third subset of said geometric entities as disjoint curves
10 when said third subset shares no common endpoints.

1 11. The method of claim 7, wherein said step of determining the base solid type
2 comprises the steps of:
3 classifying the base solid as a rotational sweep type when:
4 a) all closed curves in said plan view set are concentric
5 circles or closed curve sets substantially concentric
6 with the plan view main boundary; and
7 b) said second view set is substantially axi-symmetric about
8 its center;
9 classifying the base solid as a smooth blended solid type when said second
10 view set comprises only geometric entities that are lines whose ends
11 would coincide with the extensions of vertical or horizontal tangents
12 to the extrema of each of said closed curve sets within said plan
13 view set; and
14 classifying the base solid as an axial sweep type when neither the smooth blended
15 solid criteria nor rotational sweep criteria are met.

1 12. The method of claim 7, wherein said base solid is of the axial sweep type,
2 and said step of defining said base solid comprises the steps of:
3 establishing the main boundary of said plan view set;
4 identifying a view depth geometric entity associated with said plan view
5 main boundary in accordance with said relational constraints; and
6 defining an axial sweep base solid in accordance with said plan view main
7 boundary and said view depth geometric entity.

1 13. The method of claim 7, wherein said base solid is of the rotational sweep
2 type, and said step of defining a base solid comprises the steps of:
3 ascertaining a profile corresponding to said plan view main boundary;
4 determining the axis of symmetry; and
5 defining the base solid in accordance with the locus of points resulting from
6 a rotation of said profile about said axis of symmetry in arbitrarily
7 small increments.

14. The method of claim 7, wherein said base solid is of the smooth blended type, and said step of defining a base solid comprises the step of forming smooth blends as transitions between each successive closed curve set.

1 15. The method of claim 7, wherein said step of defining a plurality of solid
2 atoms comprises the steps of:

3 performing, for each of said view sets taken in an order determined by said ordering
4 step, the steps of:

5 a) designating said view set as the current view set;

6 b) processing each of said disjoint curves, wherein said processing
7 comprises the steps of:

8 1) establishing the orientation of said disjoint curve, wherein
9 said orientation is either vertical, horizontal, or
10 oblique;

11 2) establishing the curve type, wherein said curve type is
12 either a line or non-line;

13 3) defining a solid atom associated with said disjoint curve
14 in accordance with said curve type and orientation;

15 c) processing each of said closed curve sets, wherein said processing
16 comprises the steps of:

17 1) establishing the extrema orientation of said closed curve
18 set;

19 2) establishing the extrema curve type, wherein said extrema
20 curve type is either linear or non-linear;

21 3) establishing whether the YMAX extrema is a point-on-a-curve;

22 4) defining a solid atom associated with said closed curve
23 set in accordance with said curve type and
24 orientation;

25 c) processing each of said open curve sets, wherein said processing
26 comprises the steps of:

27 1) locating a touching curve within said current view set
28 which contacts said open curve set;

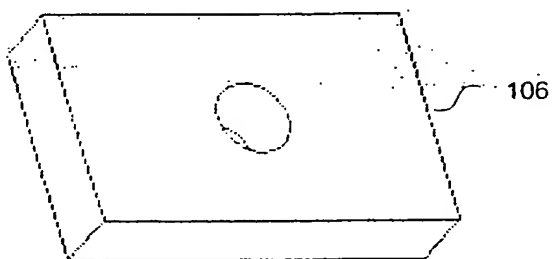
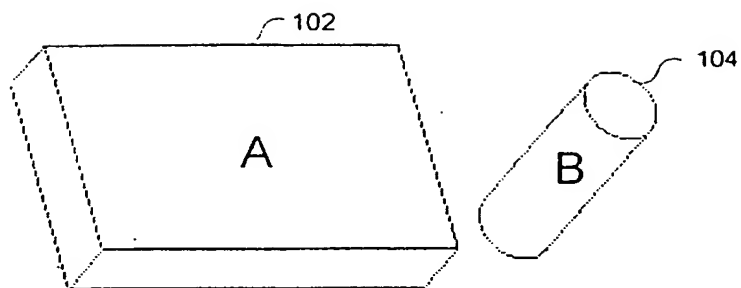
- 29 2) determining the orientation of said touching curve;
30 3) determining whether said open curve set is semi-
31 rectangular; and
32 4) defining a solid atom associated with said open curve set
33 in accordance the orientation of said touching curve
34 and whether said open curve set is semi-rectangular.

1 16. A digital computer configured to generate a three-dimensional geometric
2 solid from two-dimensional drawing data, wherein said two-dimensional drawing data
3 comprises a plurality of geometric entities, said computer comprising:
4 memory means for storing said two-dimensional drawing data;
5 a processor configured to:
6 access said memory means;
7 segregate said geometric entities into a plurality of view sets;
8 identify a plurality of relational constraints associated with
9 said geometric entities; and
10 generate said three-dimensional geometric solid in
11 accordance with said relational constraints and said
12 view sets.

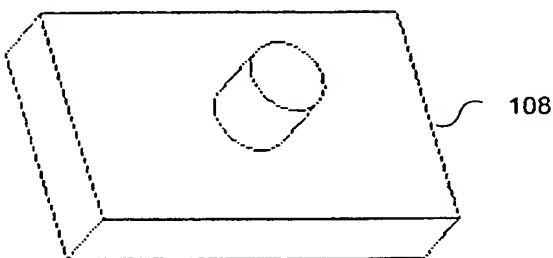
1 17. The digital computer of claim 16, further comprising:
2 a monitor for displaying said three-dimensional geometric solid;
3 a keyboard, connected to said processor, for facilitating user interaction; and
4 a printer device, connected to said processor, for printing said three-dimensional
5 geometric solid.

18. The digital computer of claim 16, further comprising:
a means for connecting said computer to a numerical control apparatus configured
to create solid objects in accordance with said three-dimensional geometric
solid.

19. The digital computer of claim 16, wherein said processor is further configured to convert said drawing data to and from a plurality of drawing standards.



$A - B$
(Difference)



$A + B$
(Union)



$A \cap B$
(Intersection)

Fig. 1A

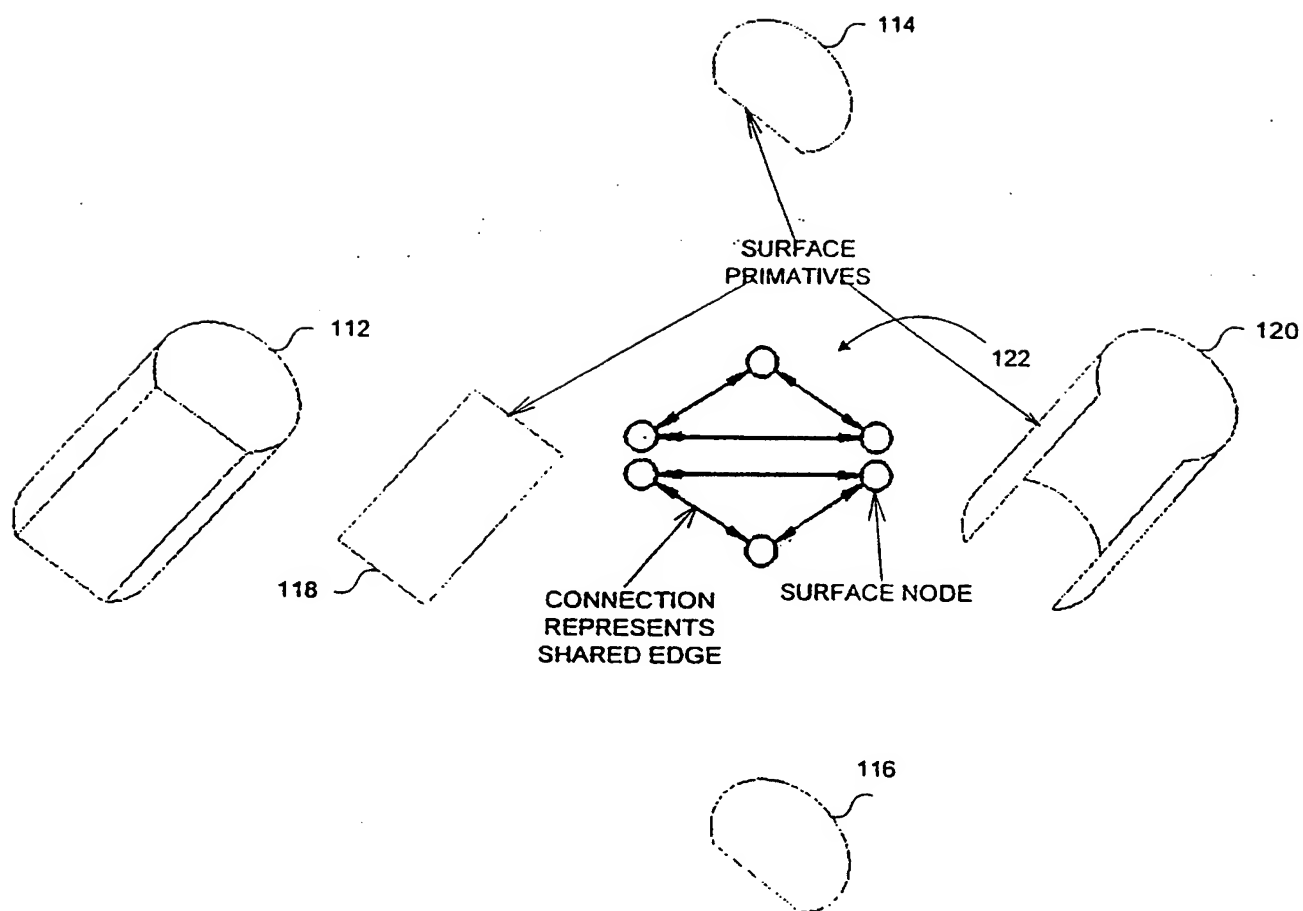


Fig. 1B

2/64

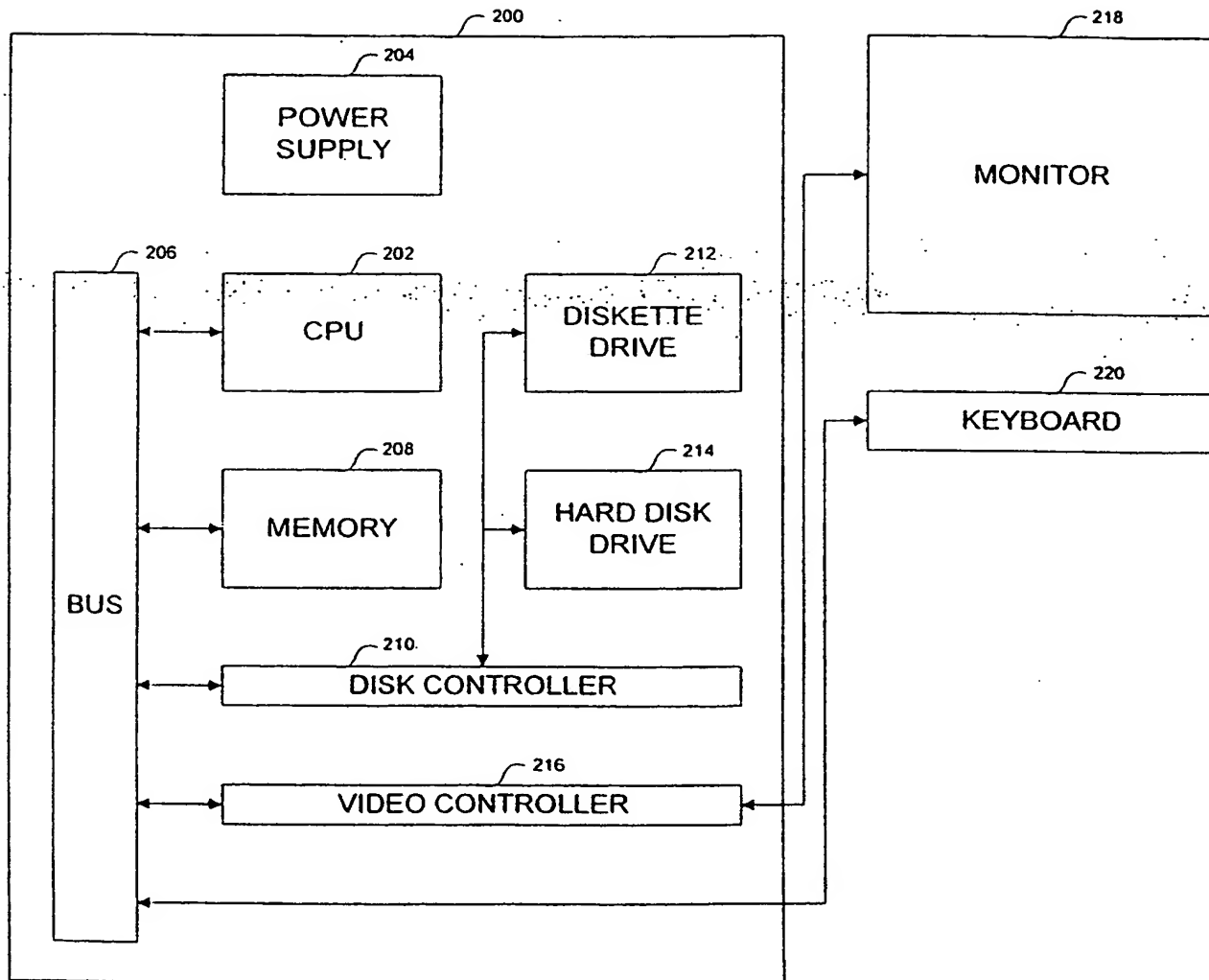


Fig. 2

3/64

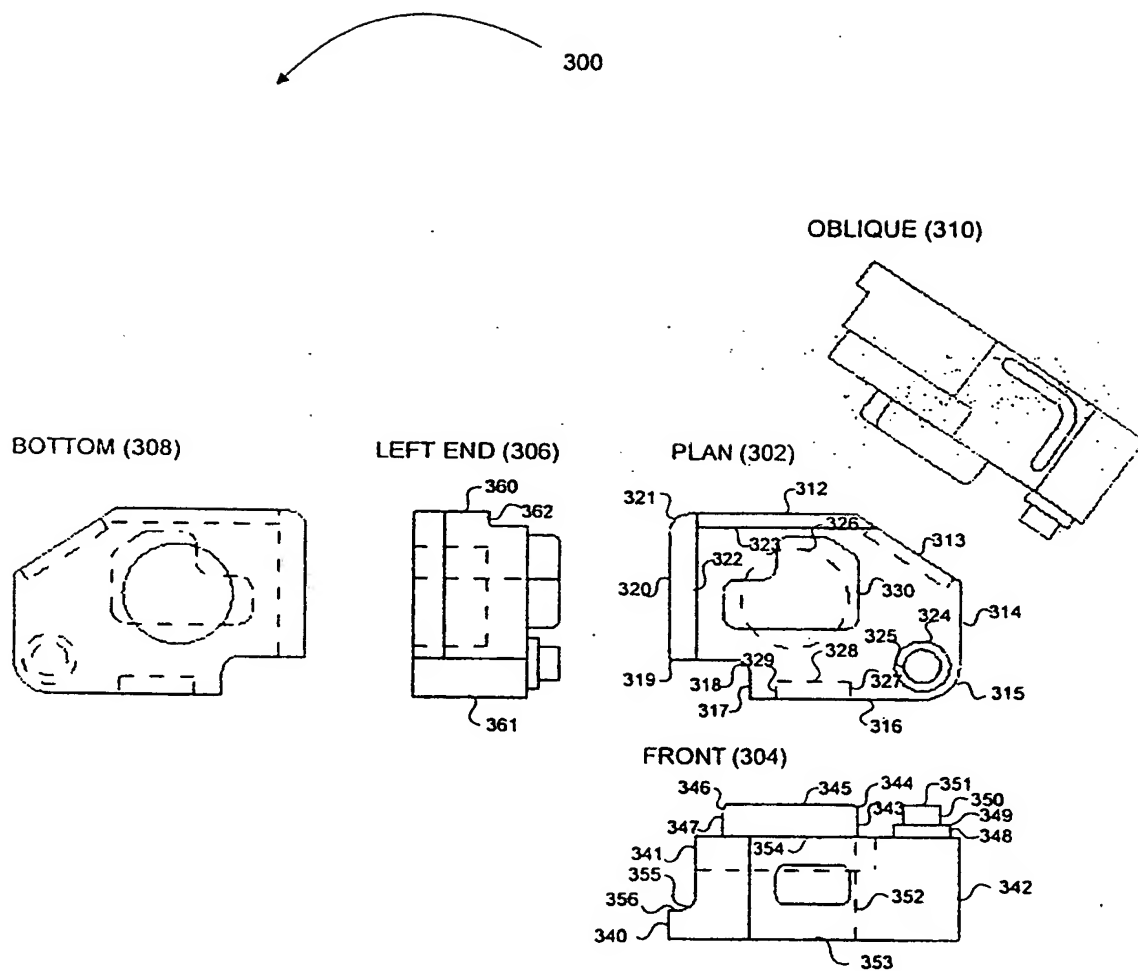


Fig. 3

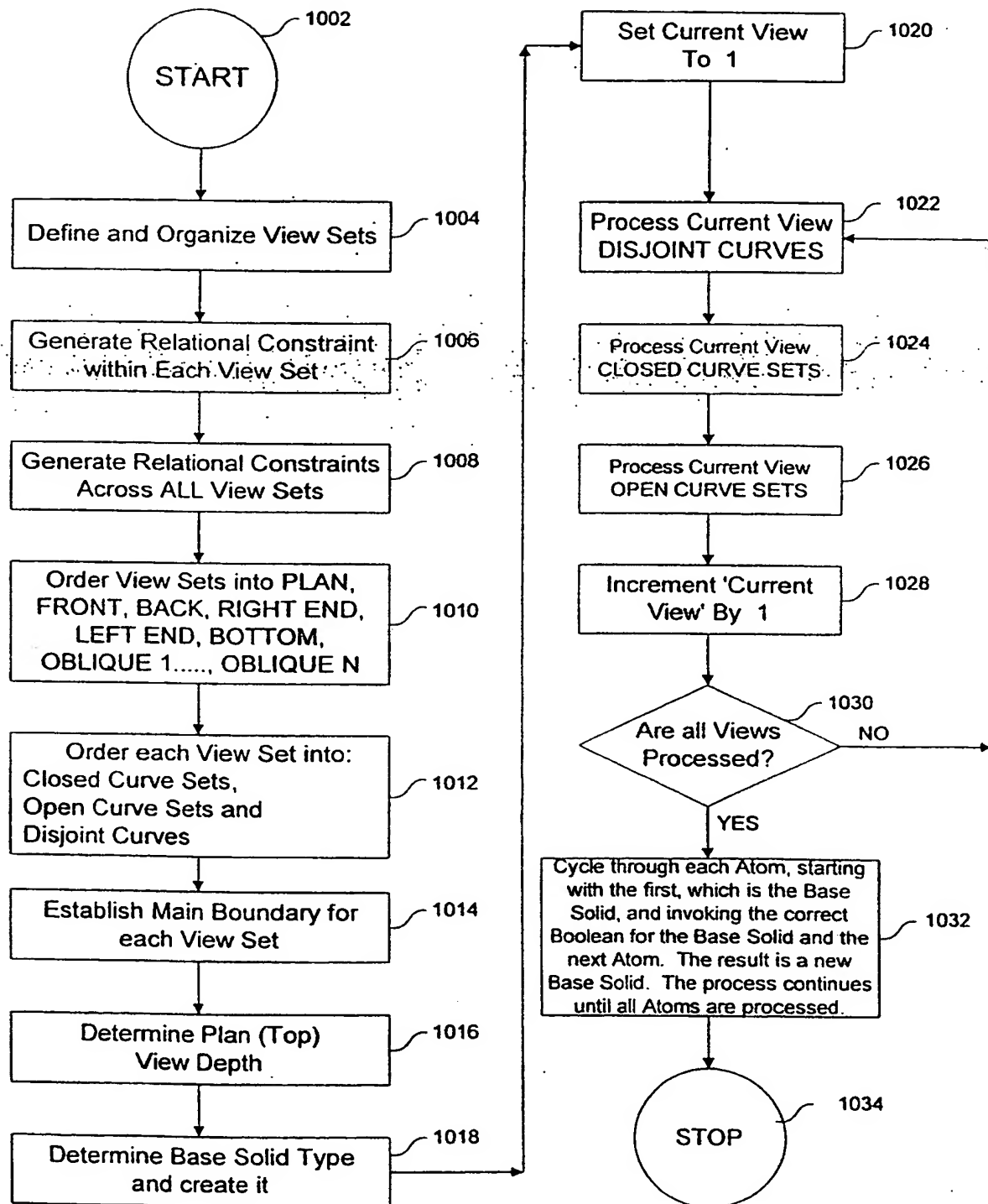


Fig. 4

5/64

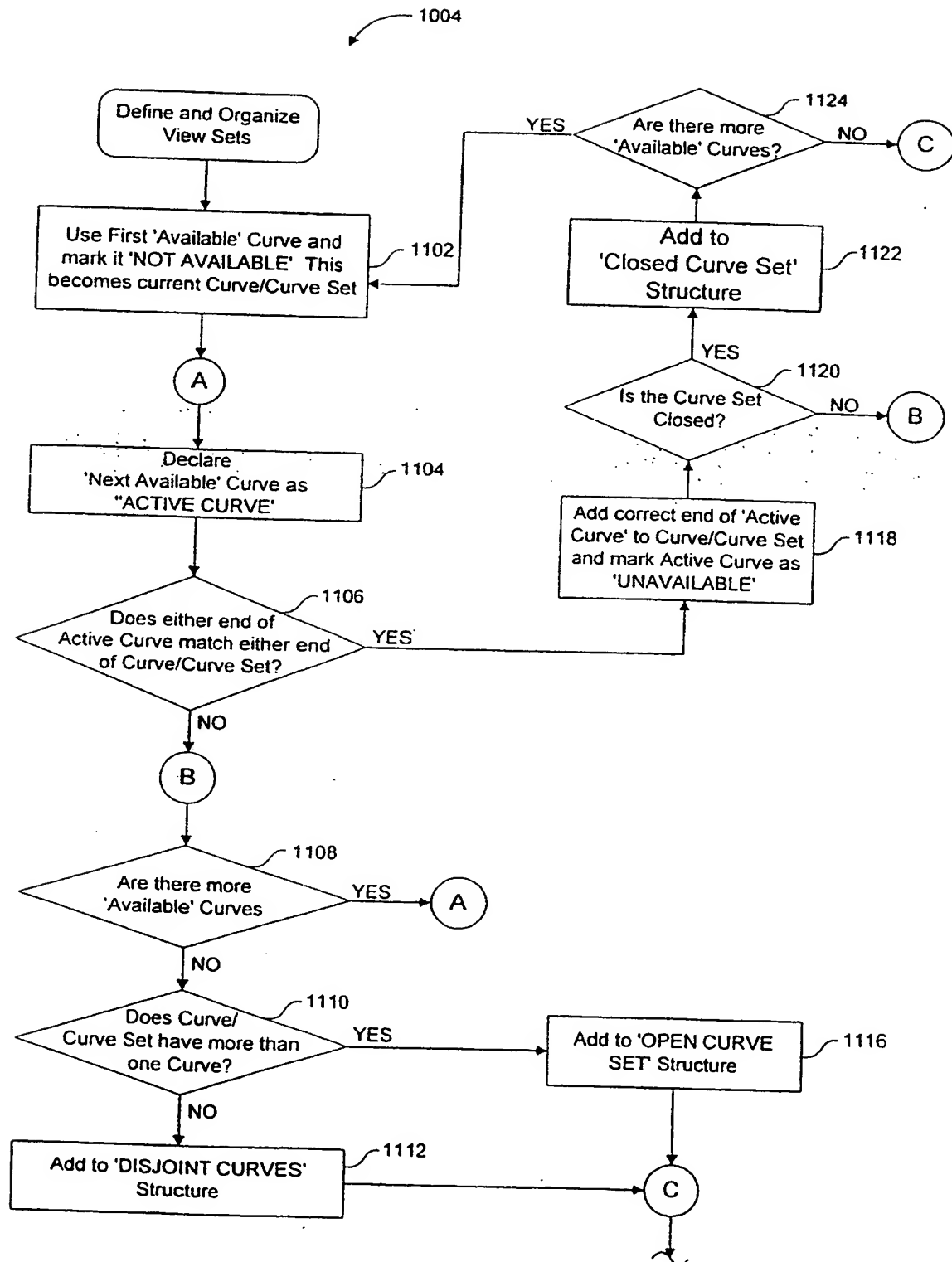
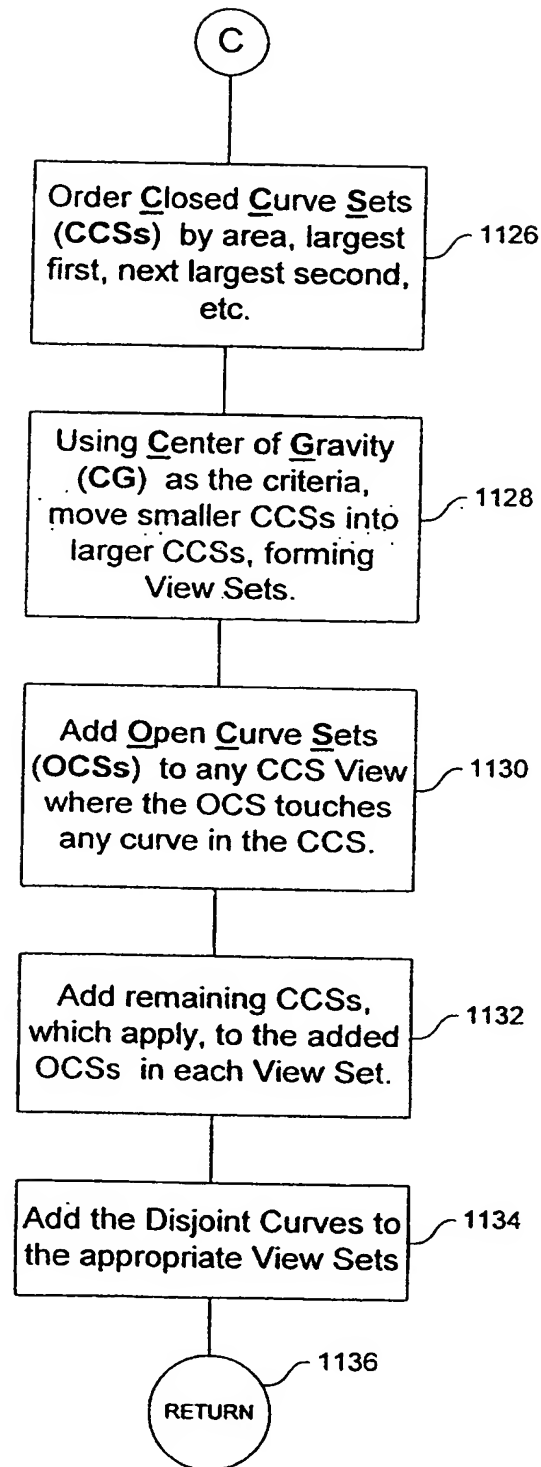


Fig. 5A

6/64

*Fig. 5B*

7/64

VIEW SET COUNT
VIEW SET 1 IDENTIFIER
VIEW SET 2 IDENTIFIER
VIEW SET n IDENTIFIER

STRUCTURE 1A
(VIEW CLUSTER)

CURVE COUNT
CURVE 1 IDENTIFIER
CURVE 2 IDENTIFIER
CURVE n IDENTIFIER

STRUCTURE 1B
(VIEW SET and
VIEW SET BOUNDARIES)

Fig. 5C

8/64

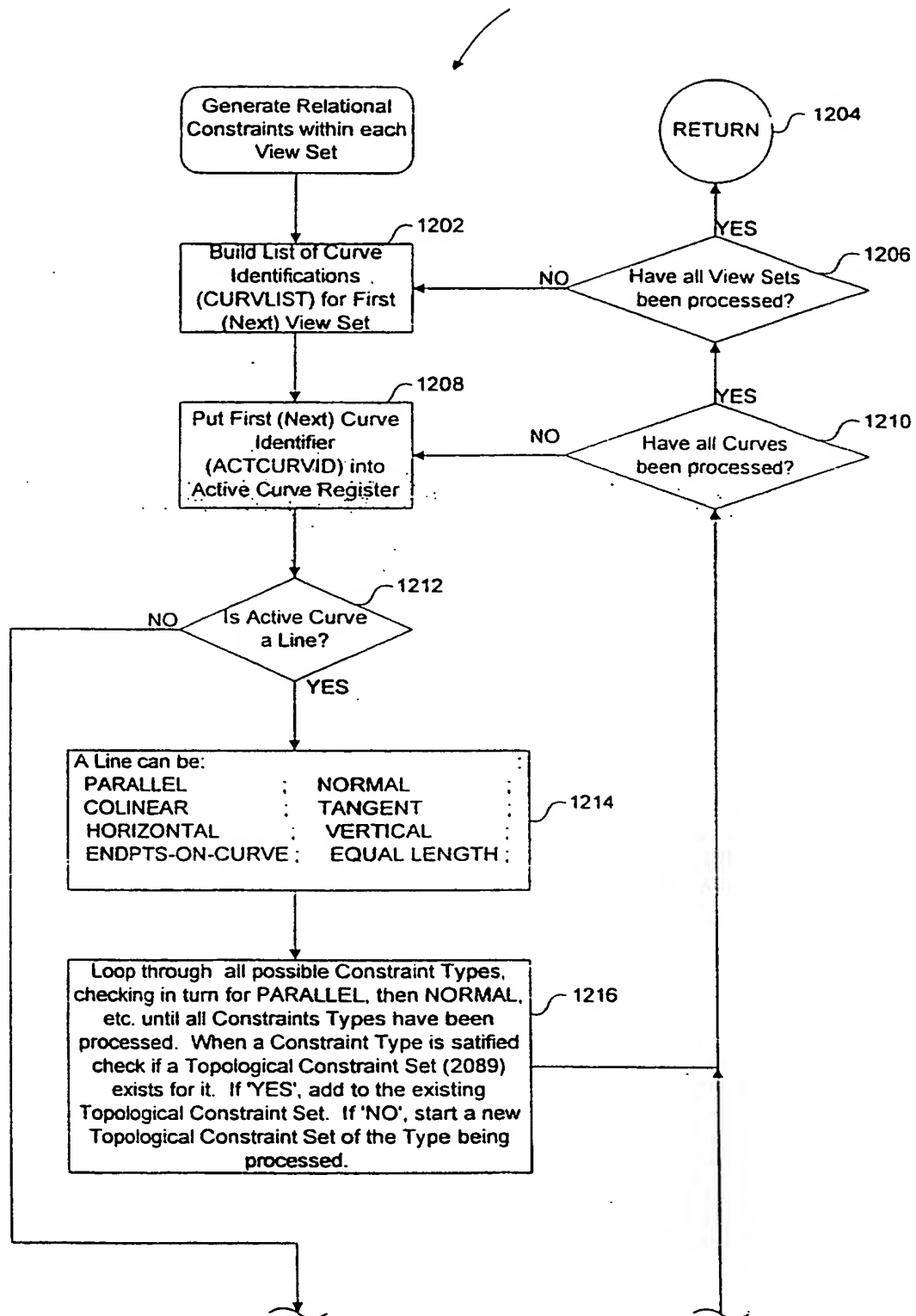


Fig. 6A

9/64

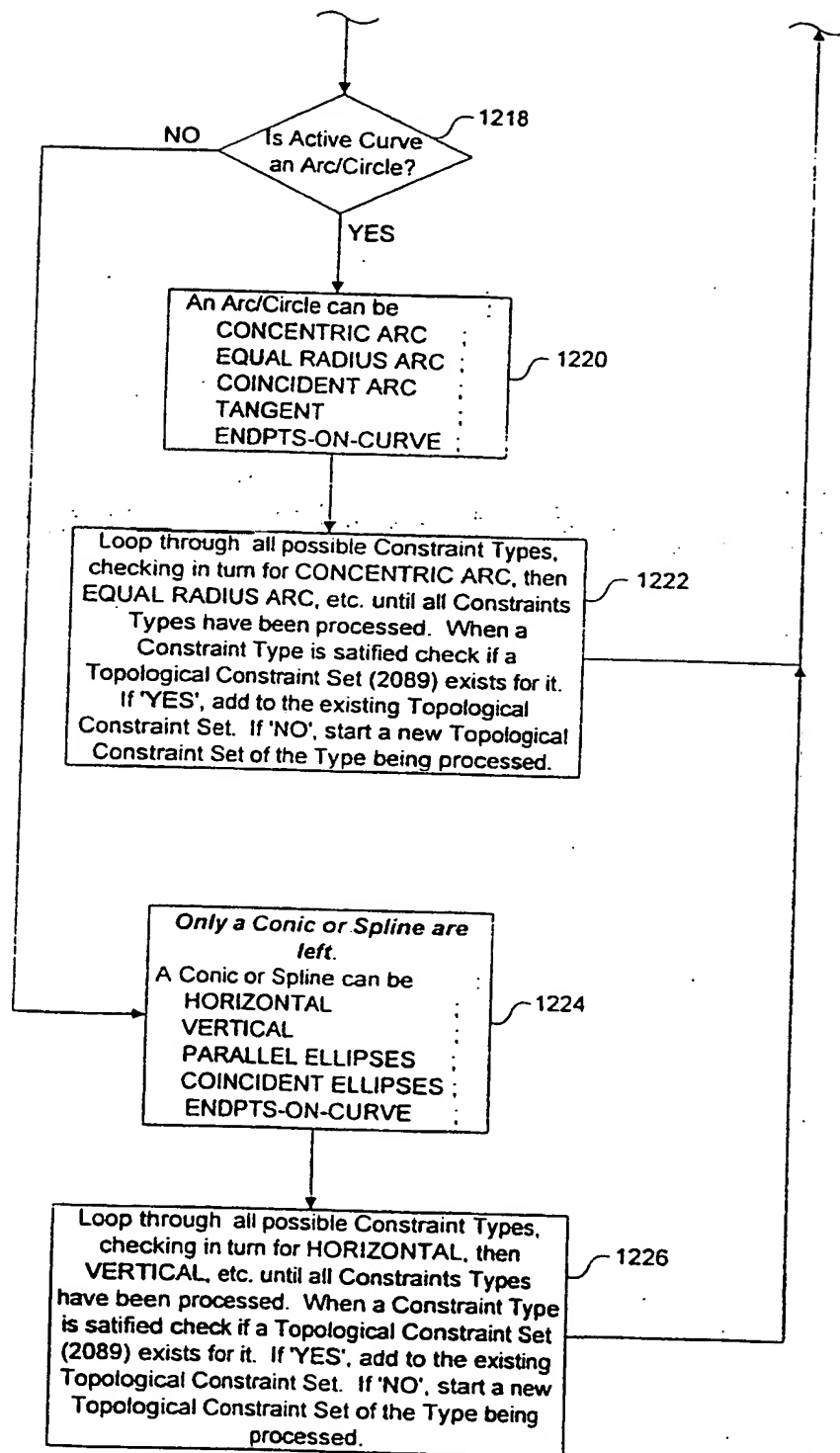


Fig. 6B

10/64

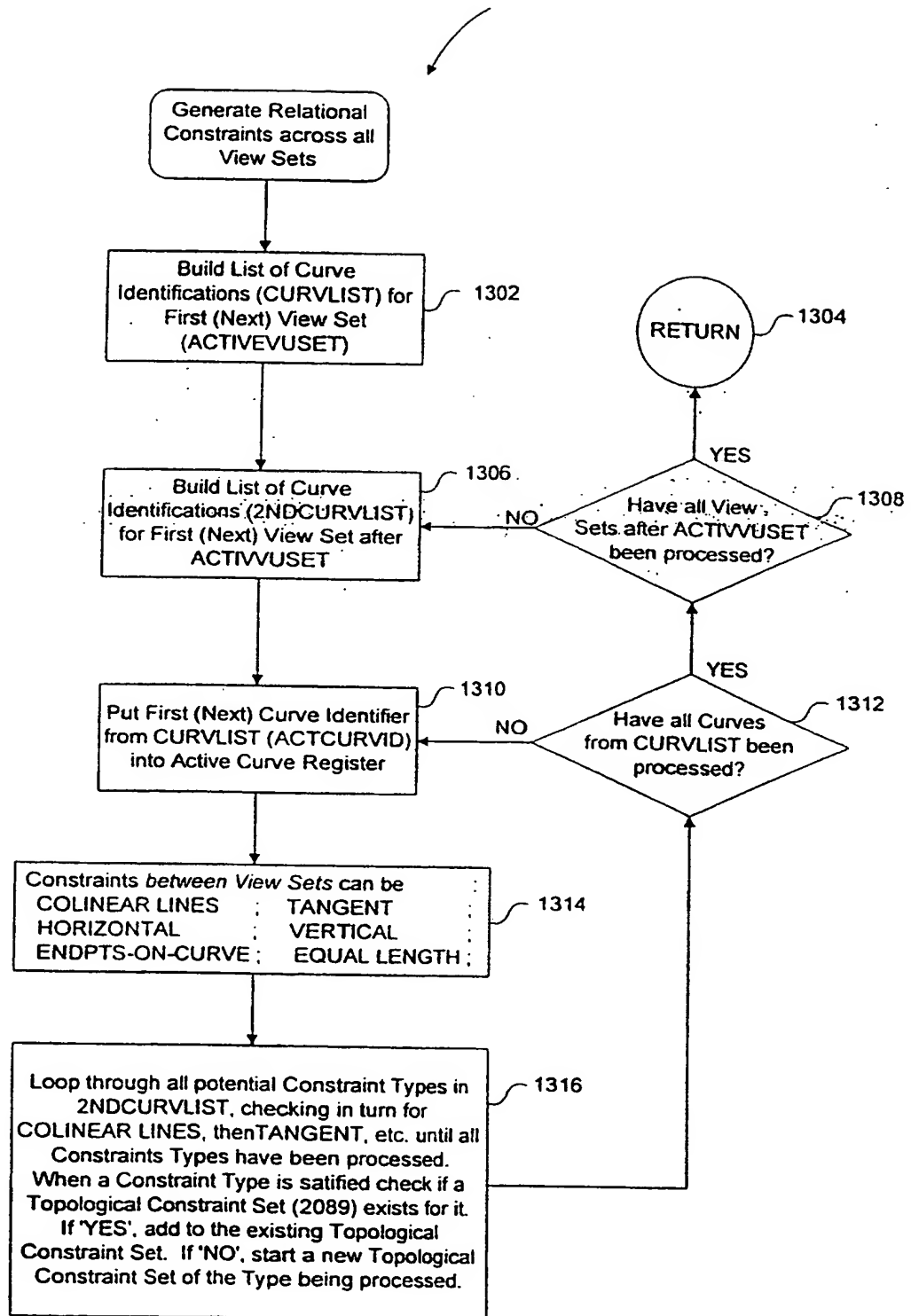


Fig. 7

11/64

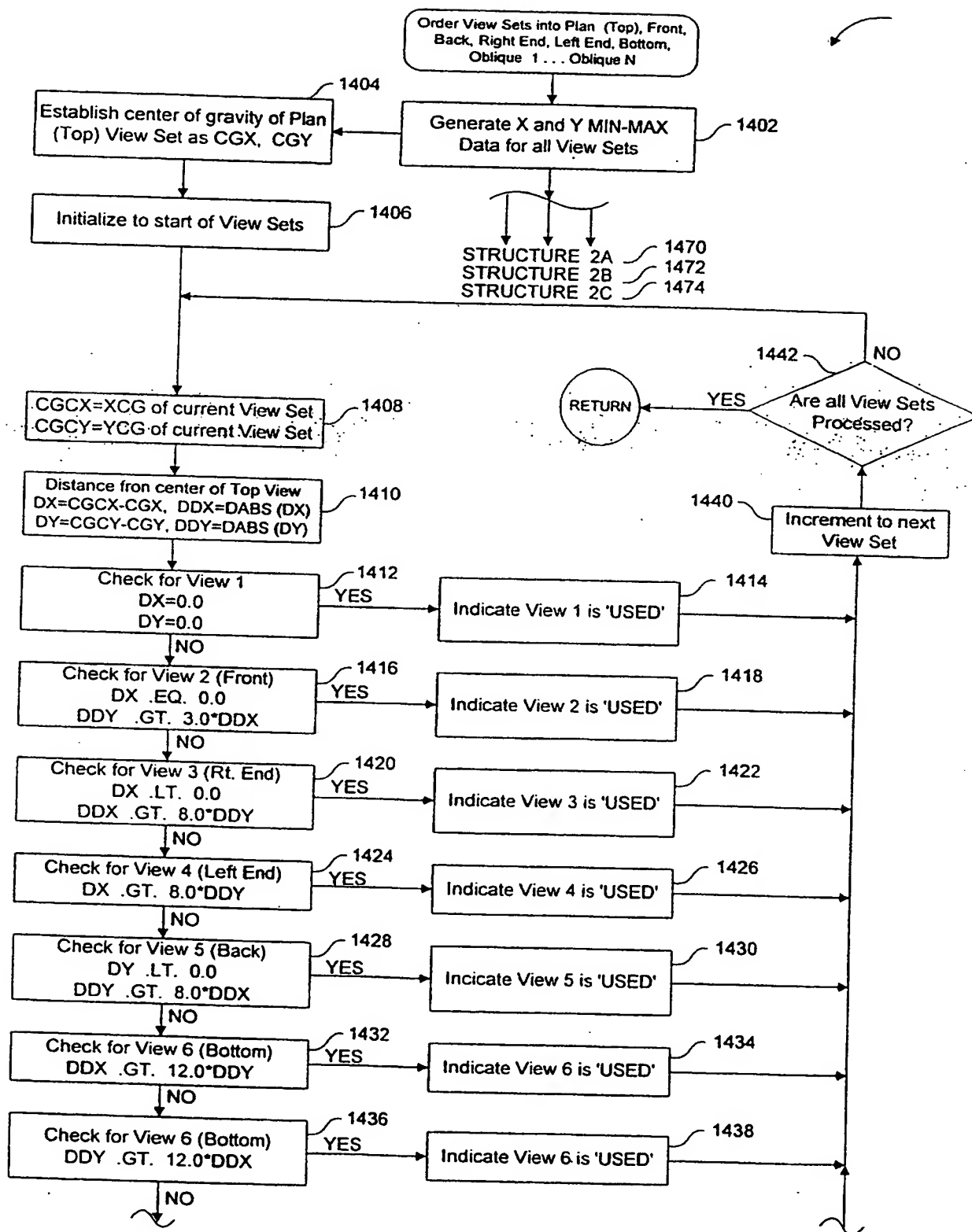


Fig. 8A

12/64

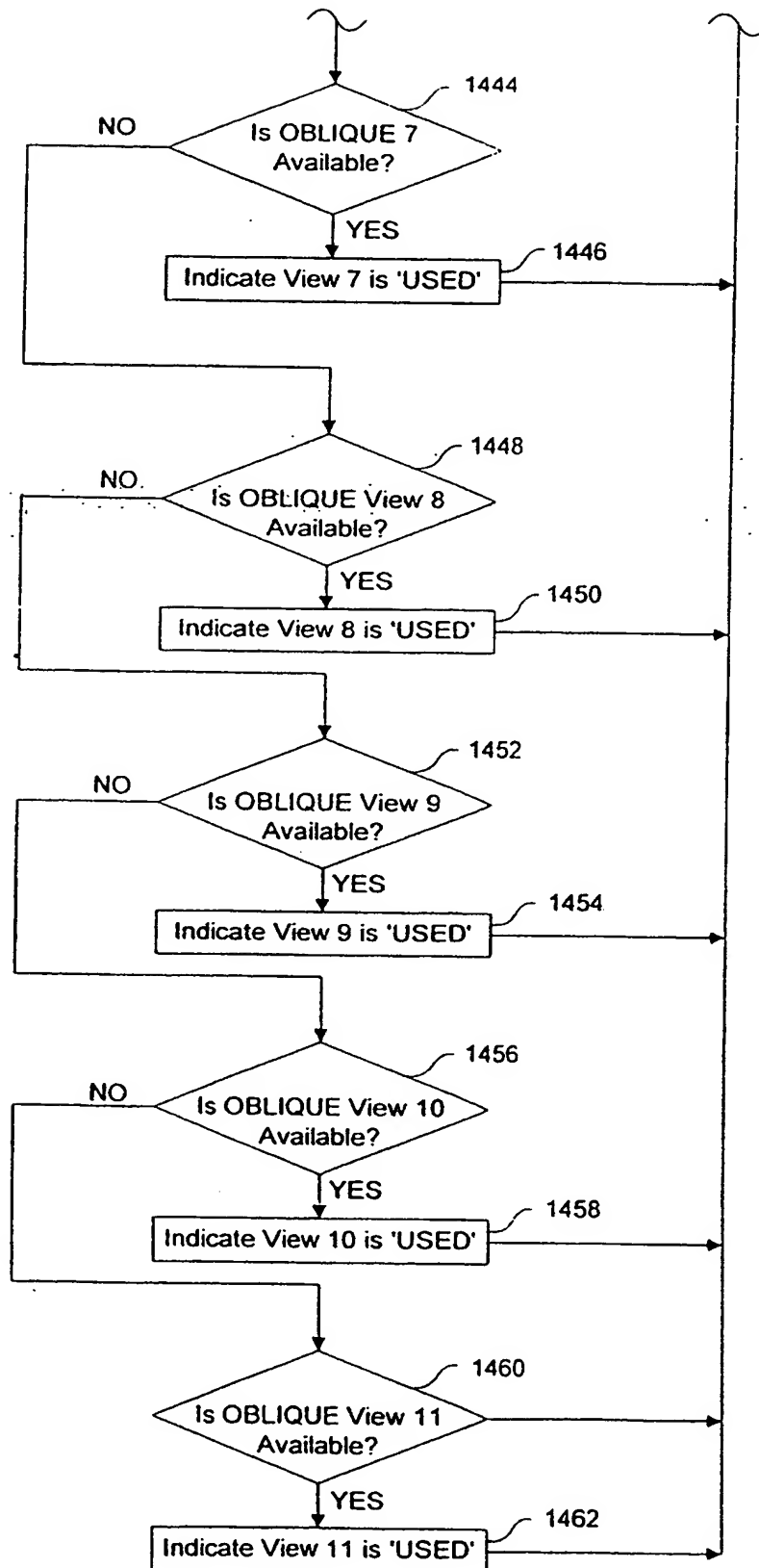


Fig. 8B

13/64

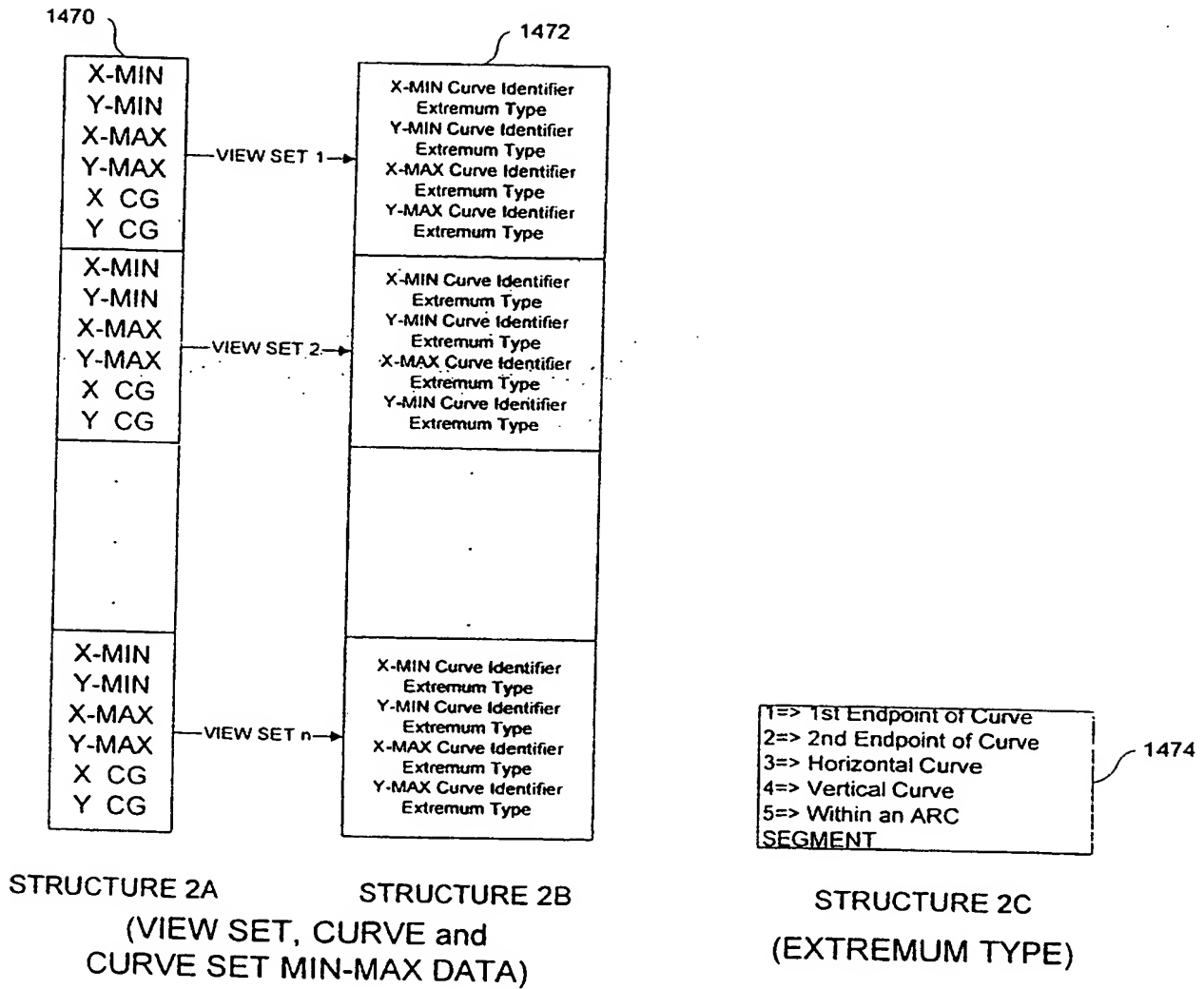


Fig. 8C

14/64

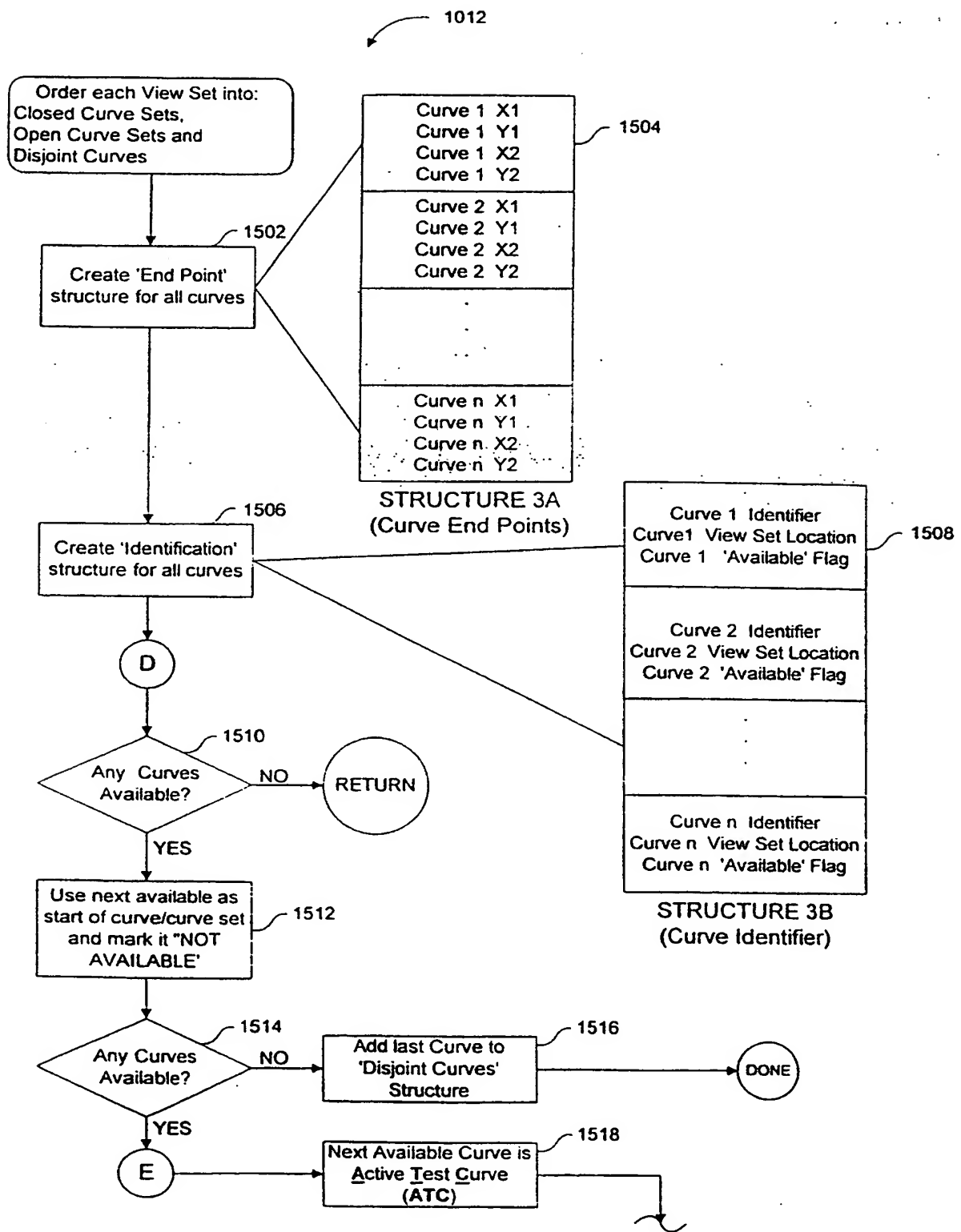


Fig. 9A

15/64

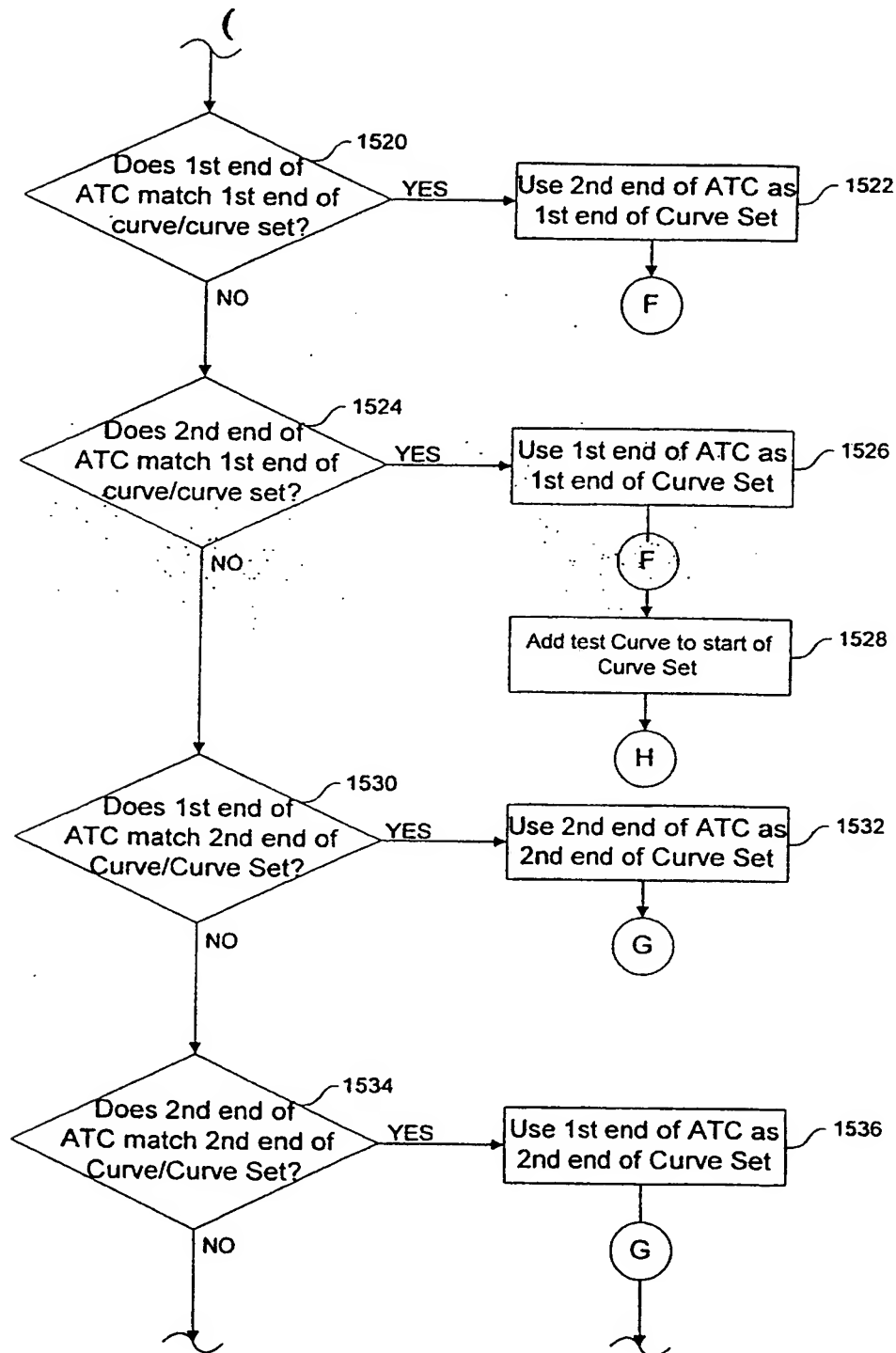


Fig. 9B

16/64

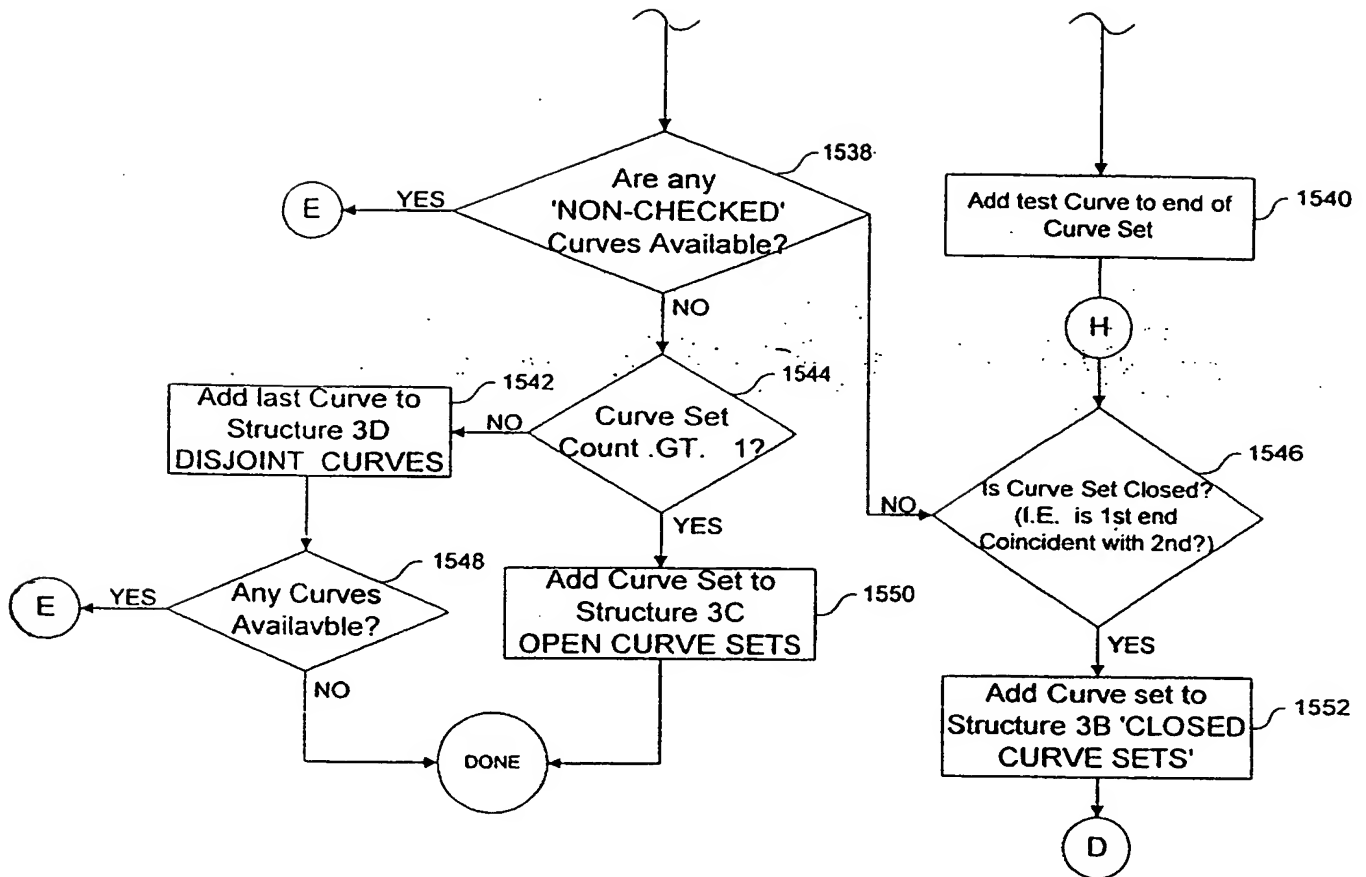
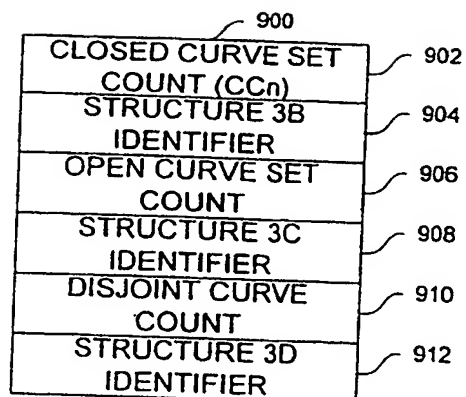
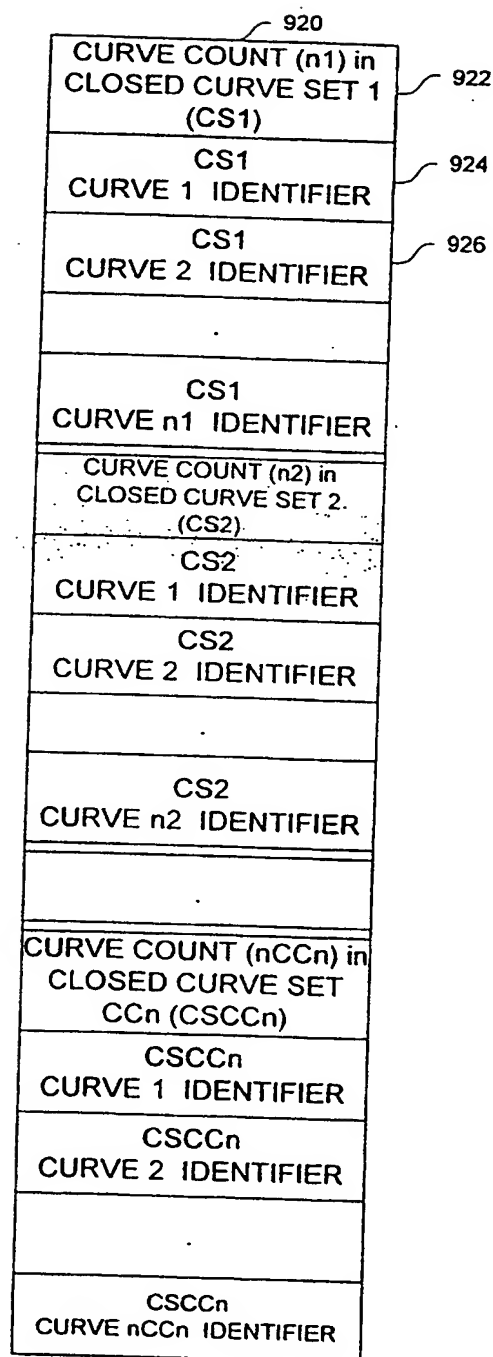


Fig. 9C

17/64



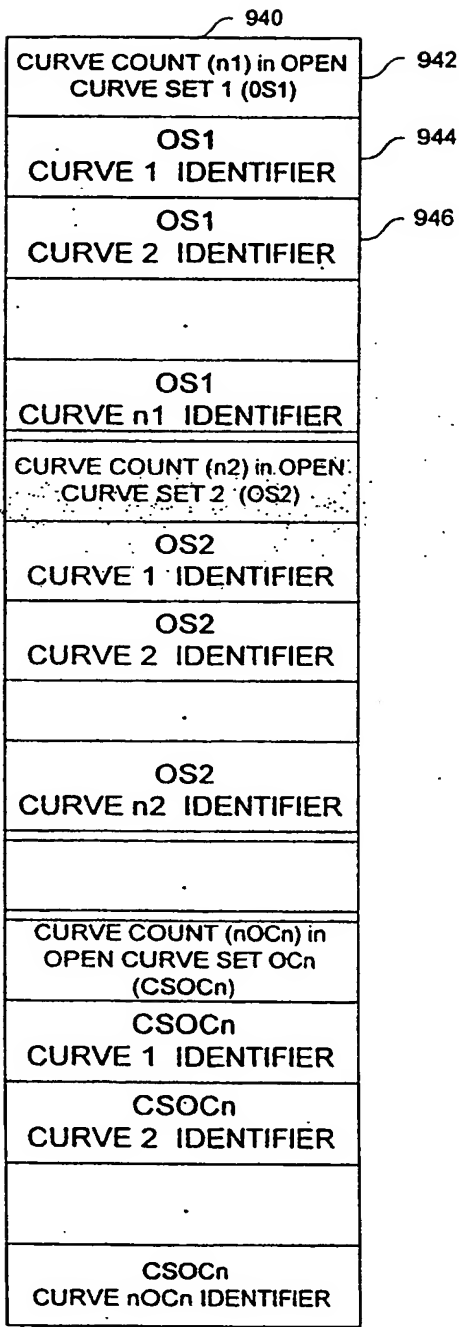
STRUCTURE 4A
(CURVE TOPOLOGY)



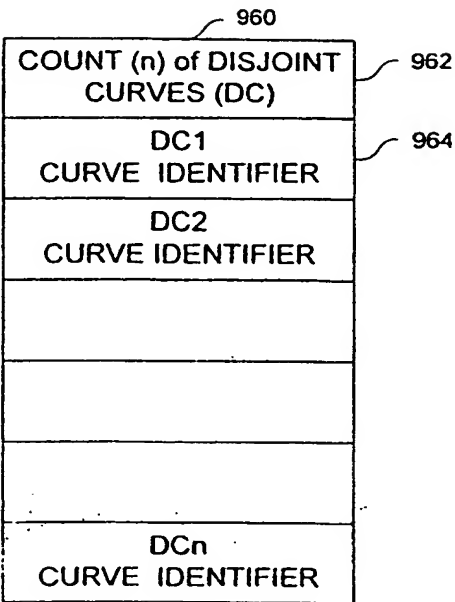
STRUCTURE 4B
(CLOSED CURVE SETS)

Fig. 9D

18/64



STRUCTURE 4C
(OPEN CURVE SETS)



STRUCTURE 4D
(DISJOINT CURVES)

Fig. 9E

19/64

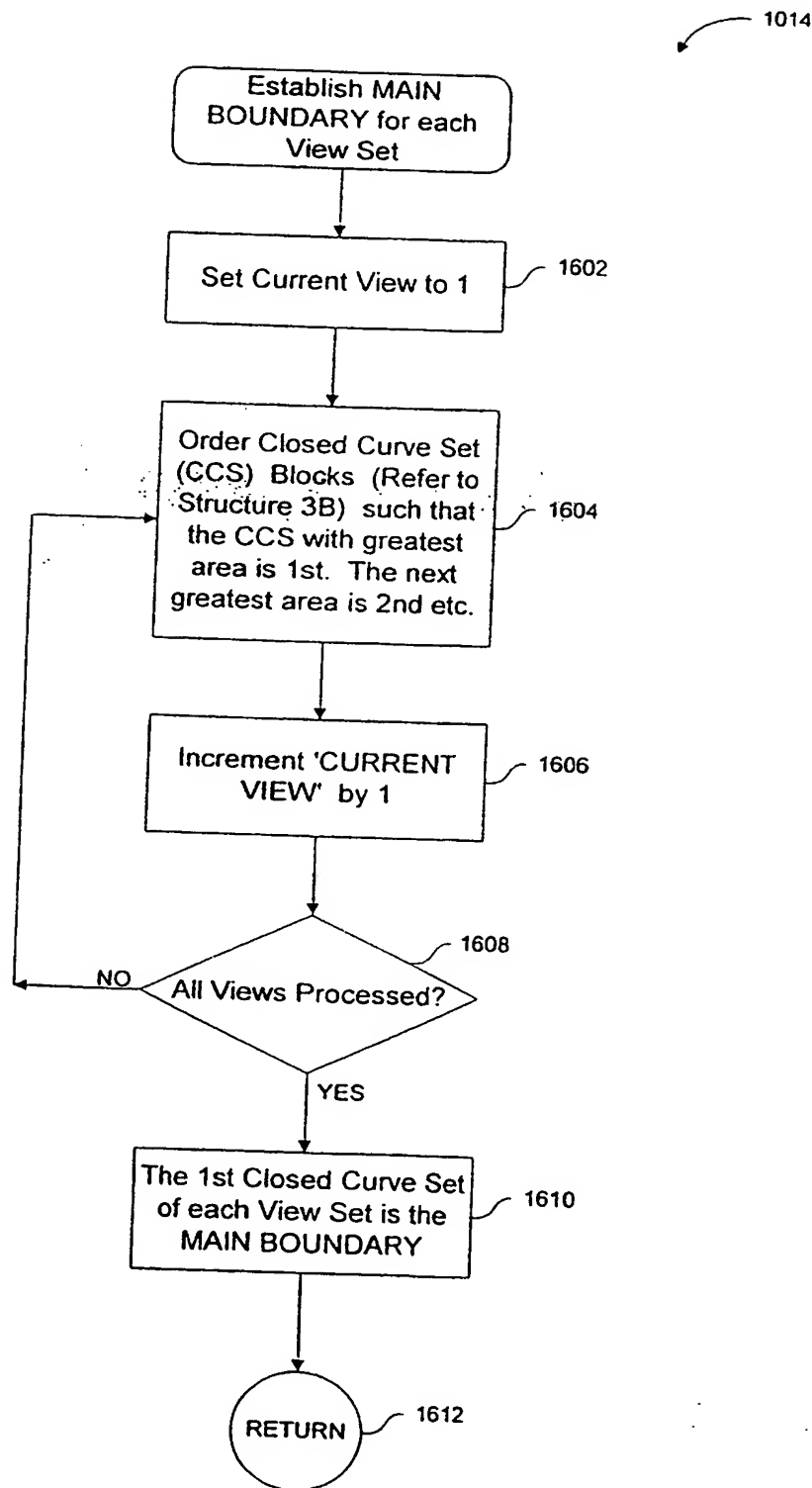


Fig. 10

20/64

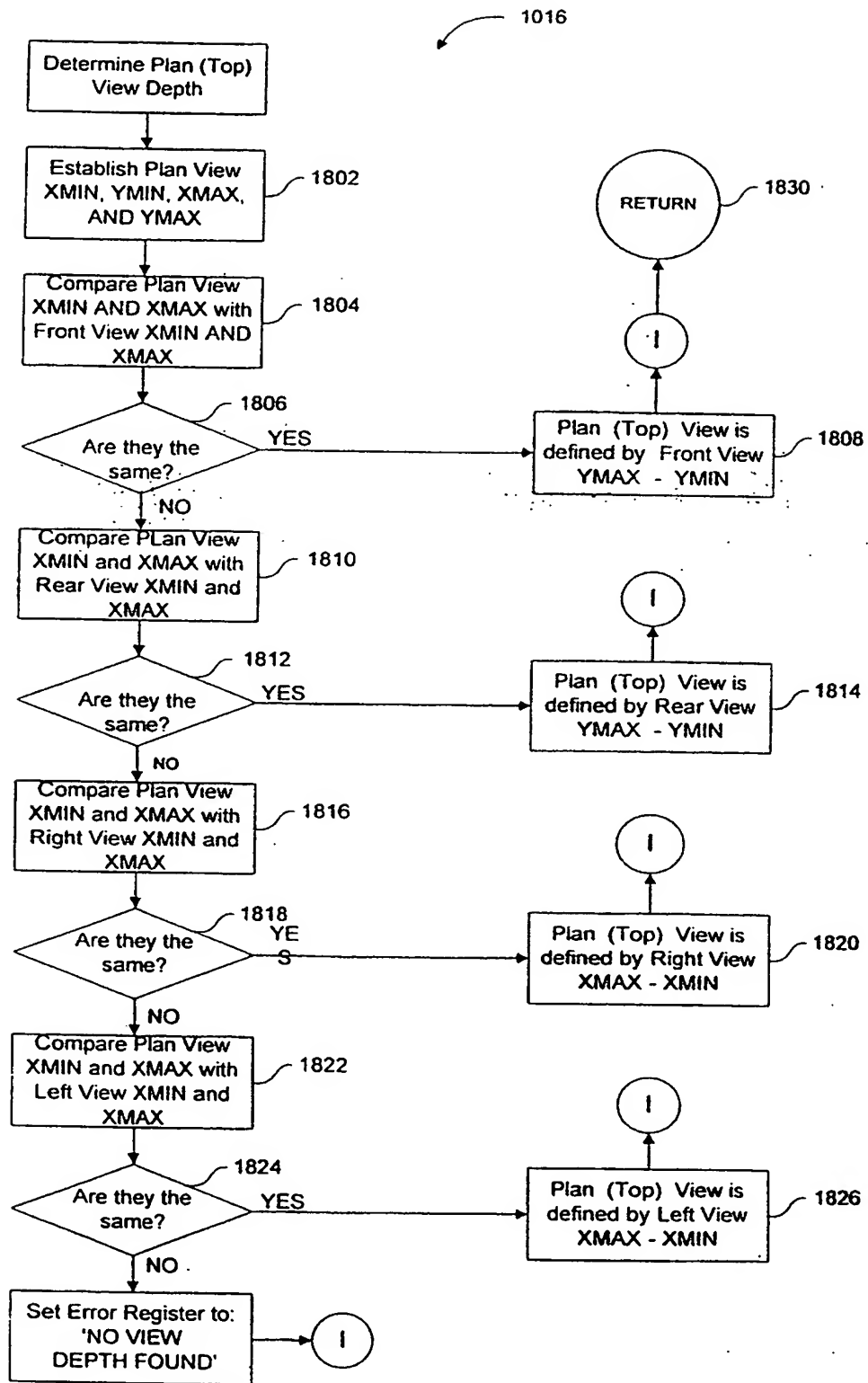


Fig. 11

21/64

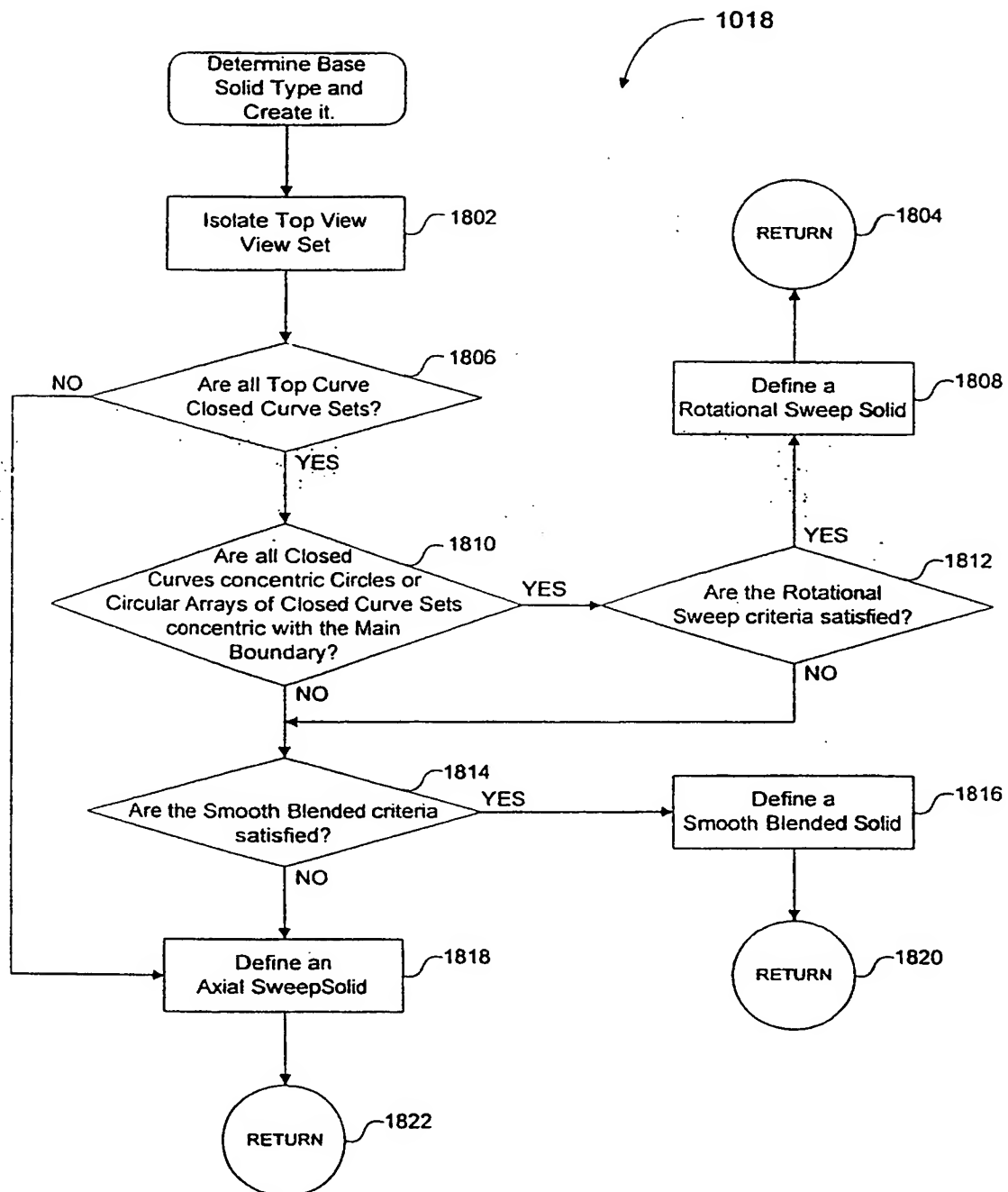


Fig. 12A

22/64

Count of Curves in Structure (nC)
Curve Identifier 1 (Axis for Type 2 only)
Curve Identifier 2
Curve Identifier nC

STRUCTURE 5A
(CURVE IDENTIFIERS)
Axial and Rotational

Count of Segments in Structure (nS)
Closed Count (n) in Segment 1 Curve 1 ID, Curve 2 ID, ..., Curve n ID View 2 Horizontal Line ID
Closed Count (n) in Segment 2 Curve 1 ID, Curve 2 ID, ..., Curve n ID View 2 Horizontal Line ID
Closed Count (n) in Segment nS Curve 1 ID, Curve 2 ID, ..., Curve n ID View 2 Horizontal Line ID

STRUCTURE 5B
(CURVE IDENTIFIERS)
Smooth Blended

Number of ATOMS (nA)	1822
Identifier to curves for Atom 1 BOOLEAN key: 1=>Union, 2=>Difference, 3=>Intersection Work View number Solid Type: 1=>Axial Sweep, 2=>Rotational Sweep, 3=>Smooth Blended ZTOP for Atom 1, ZBOTTOM for Atom 1 ZTOP Blend Radius for Atom 1, ZBOTTOM Blend Radius for Atom 1	1824
Identifier to curves for Atom 2 BOOLEAN key: 1=>Union, 2=>Difference, 3=>Intersection Work View number Solid Type: 1=>Axial Sweep, 2=>Rotational Sweep, 3=>Smooth Blended ZTOP for Atom 2, ZBOTTOM for Atom 2 ZTOP Blend Radius for Atom 2, ZBOTTOM Blend Radius for Atom 2	1826A
	1826B
Identifier to curves for Atom nA BOOLEAN key: 1=>Union, 2=>Difference, 3=>Intersection Work View number Solid Type: 1=>Axial Sweep, 2=>Rotational Sweep, 3=>Smooth Blended ZTOP for Atom nA, ZBOTTOM for Atom nA ZTOP Blend Radius for Atom nA, ZBOTTOM Blend Radius for Atom nA	1826n

STRUCTURE 5C
(ATOM DEFINITIONS)

Fig. 12B

23/64

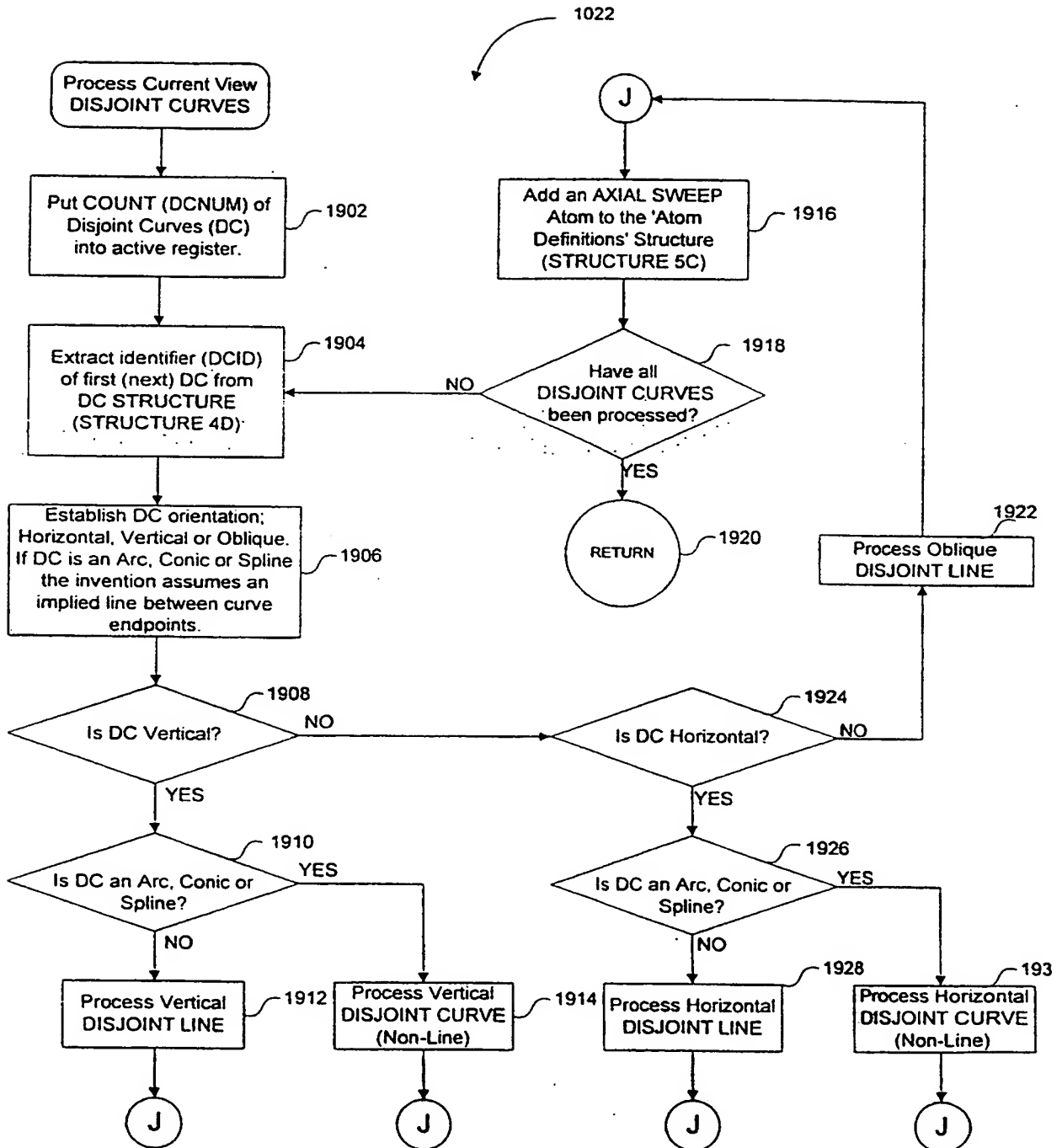


Fig. 13A

24/64

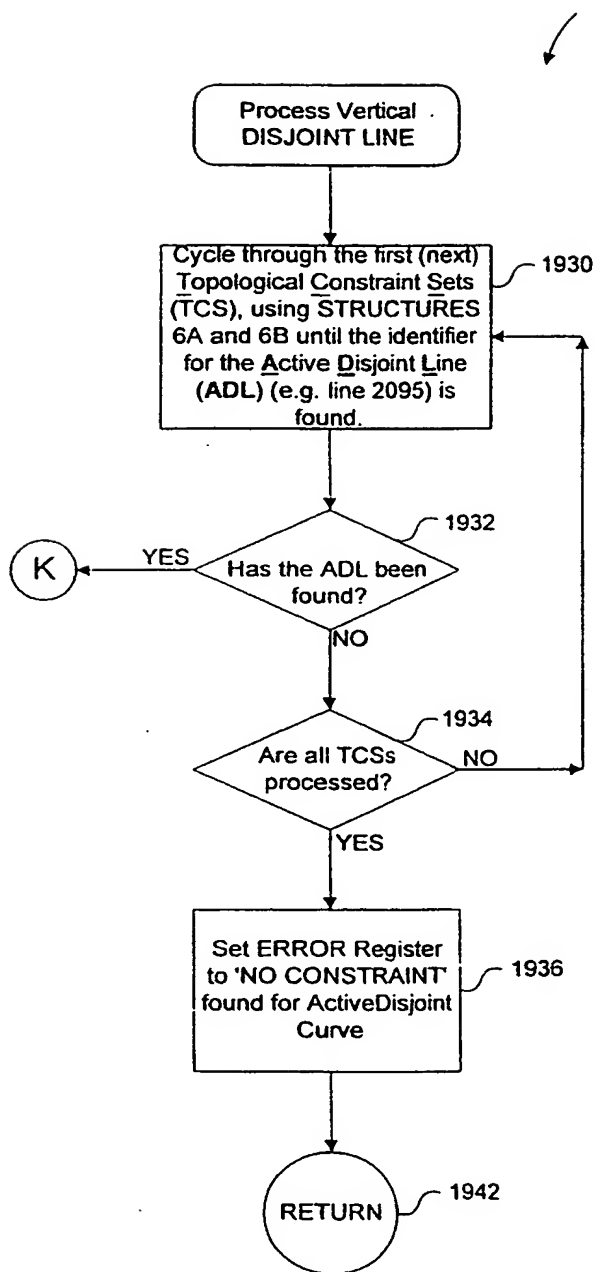


Fig. 13B

25/64

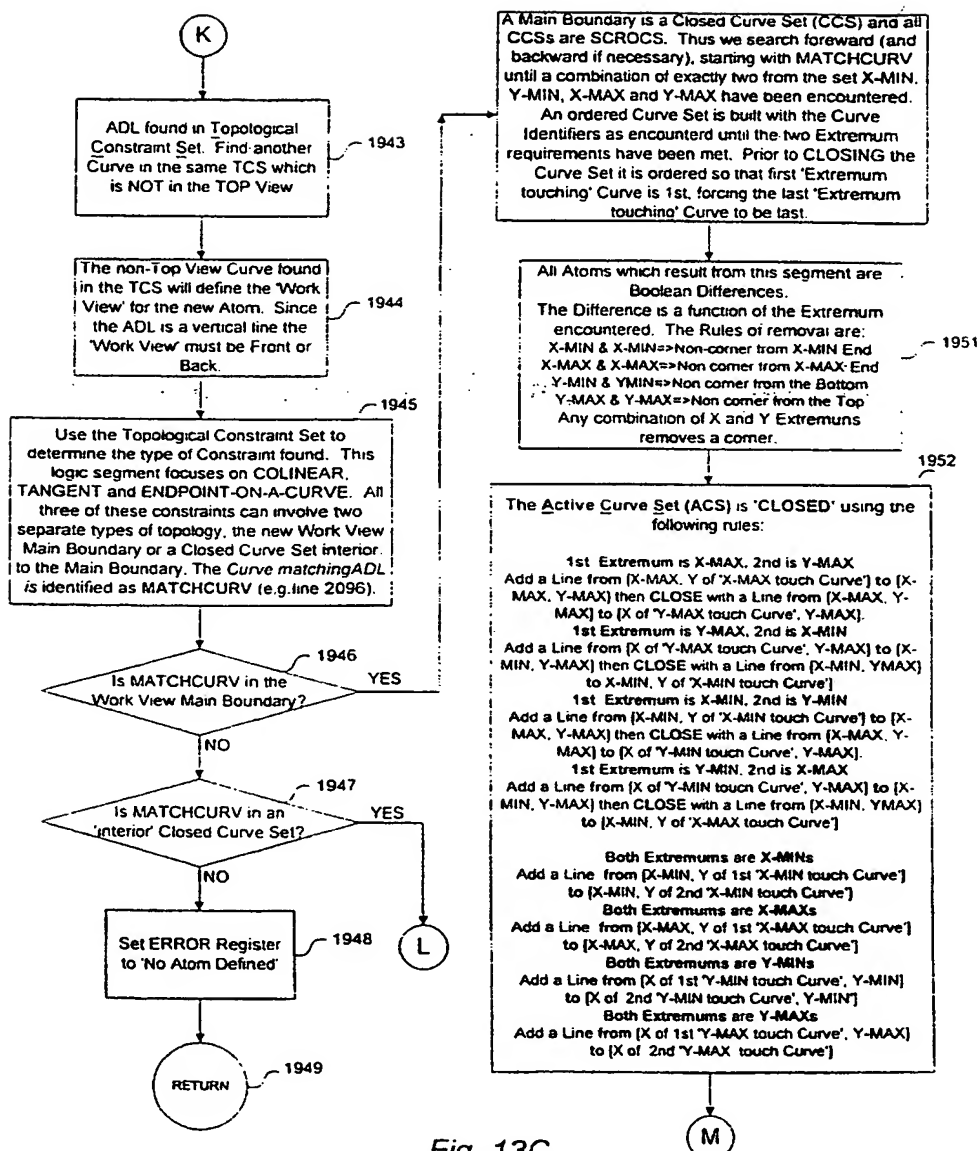


Fig. 13C

26/64

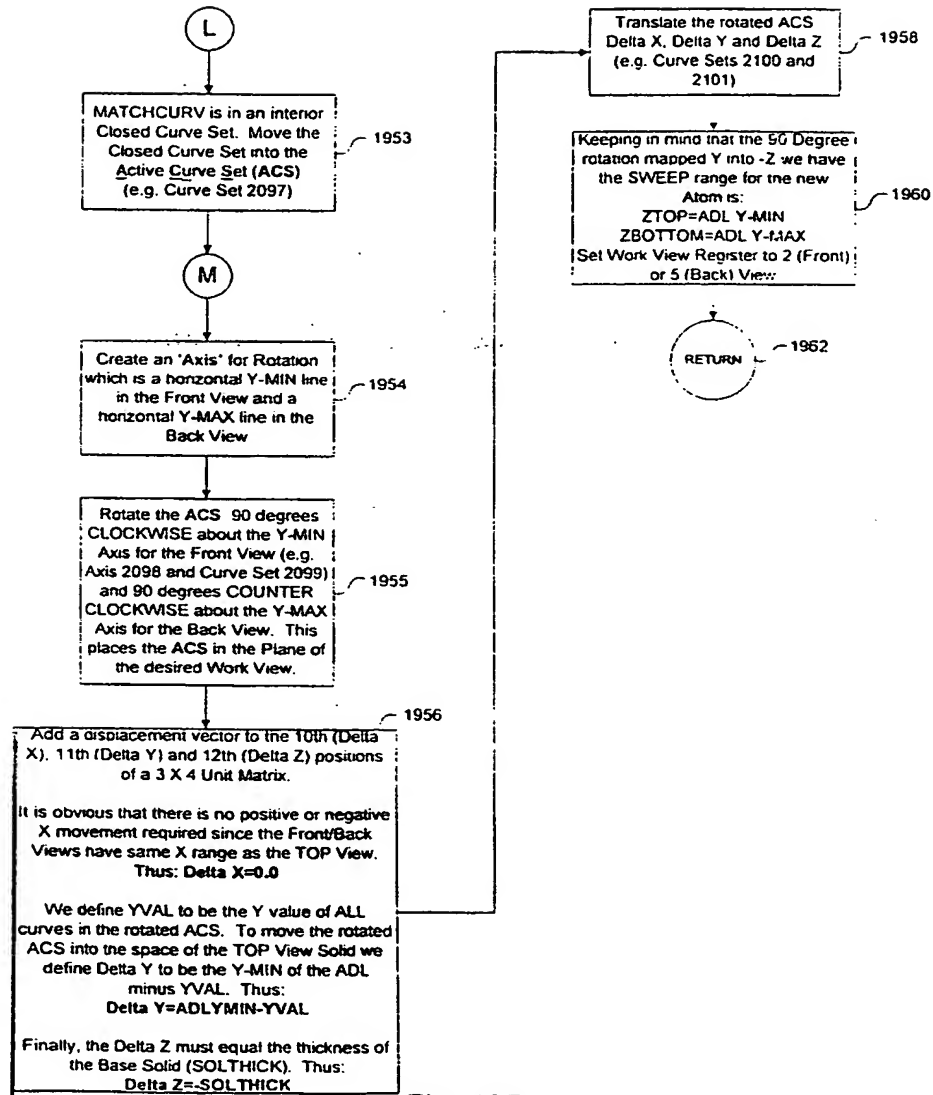


Fig. 13D

27/64

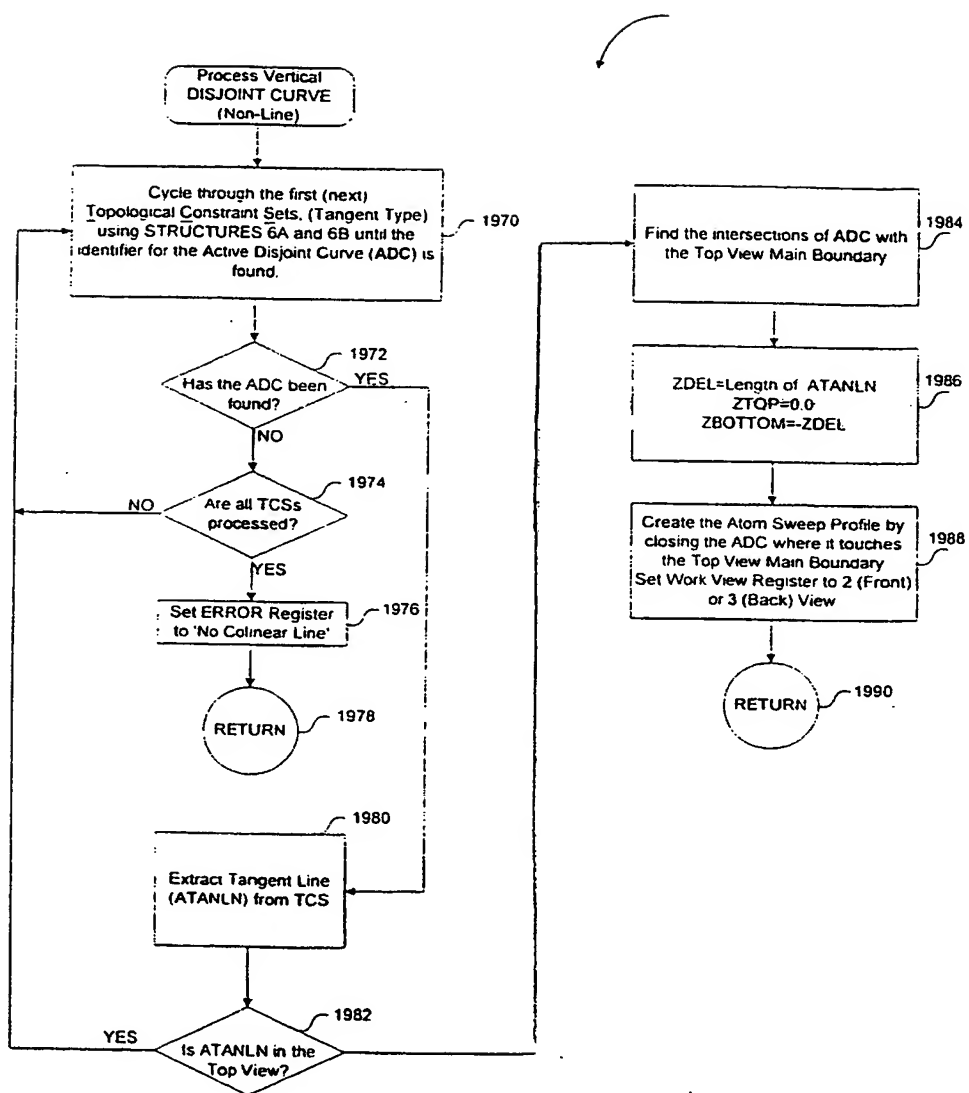


Fig. 13E

28/64

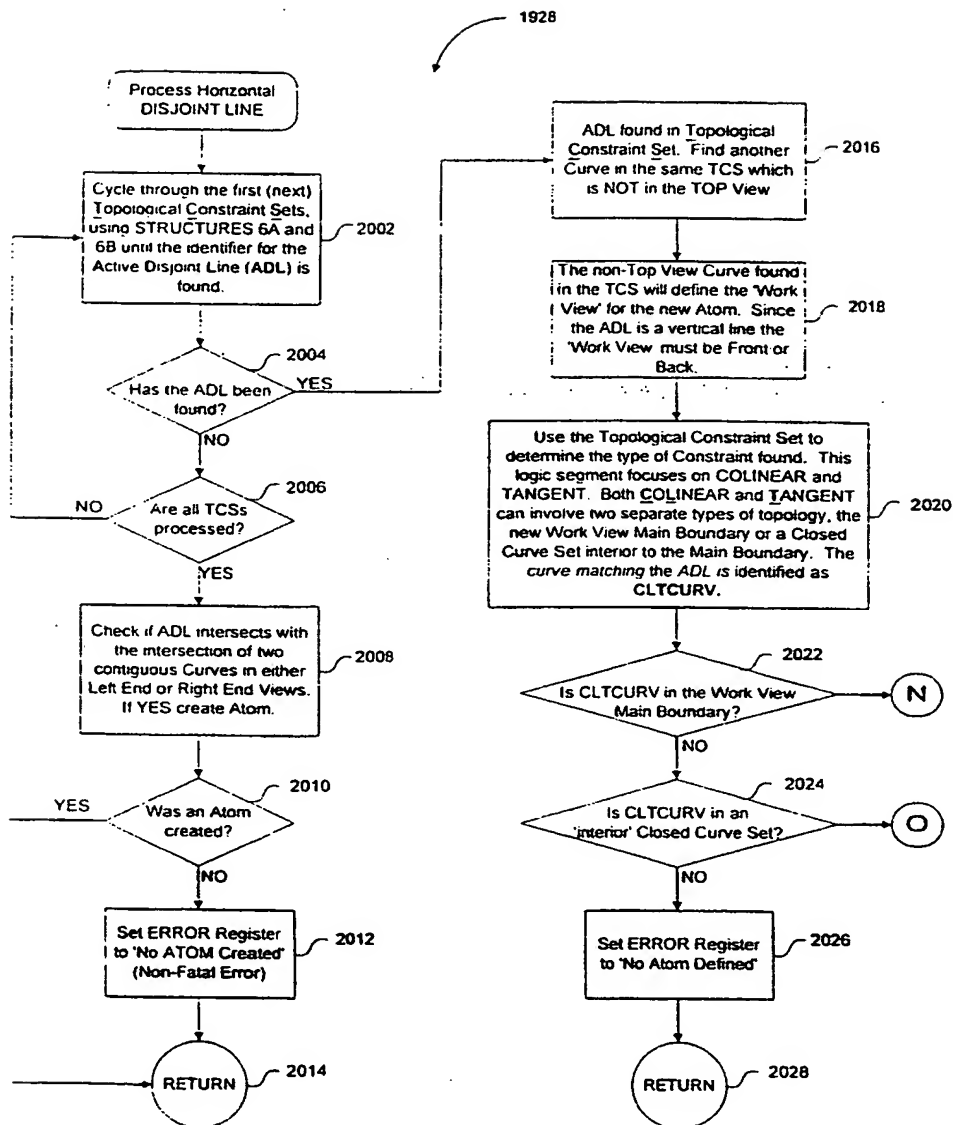


Fig. 13F

29/64

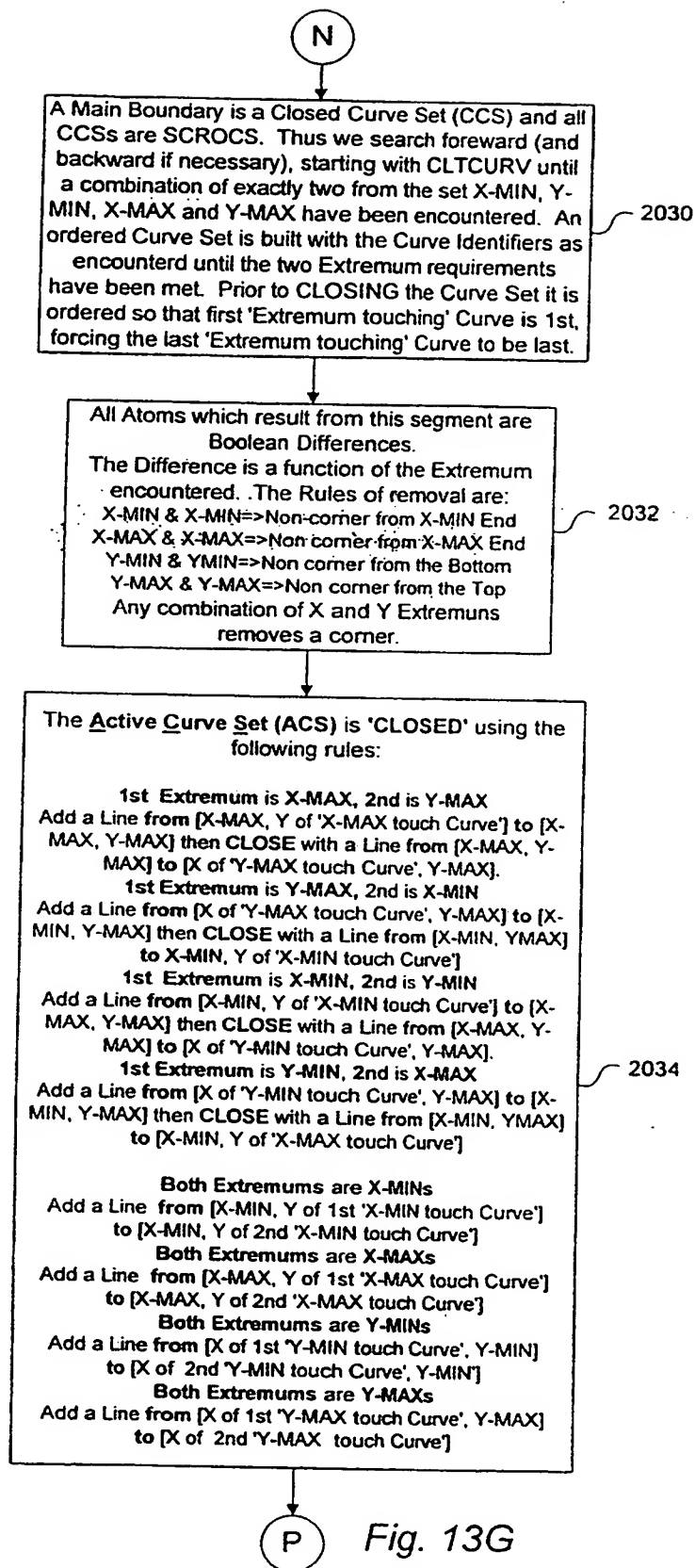


Fig. 13G

30/64

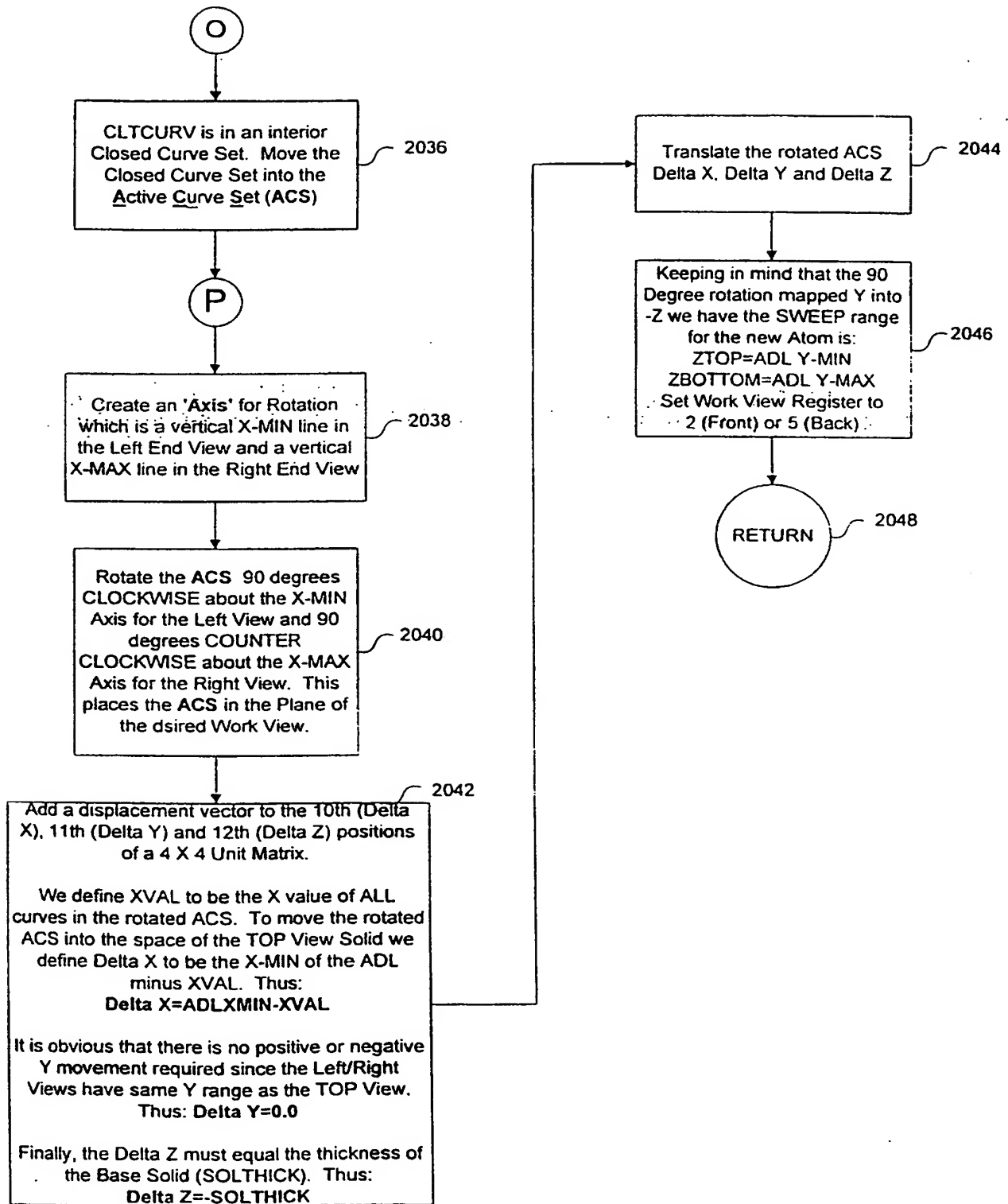


Fig. 13H

31/64

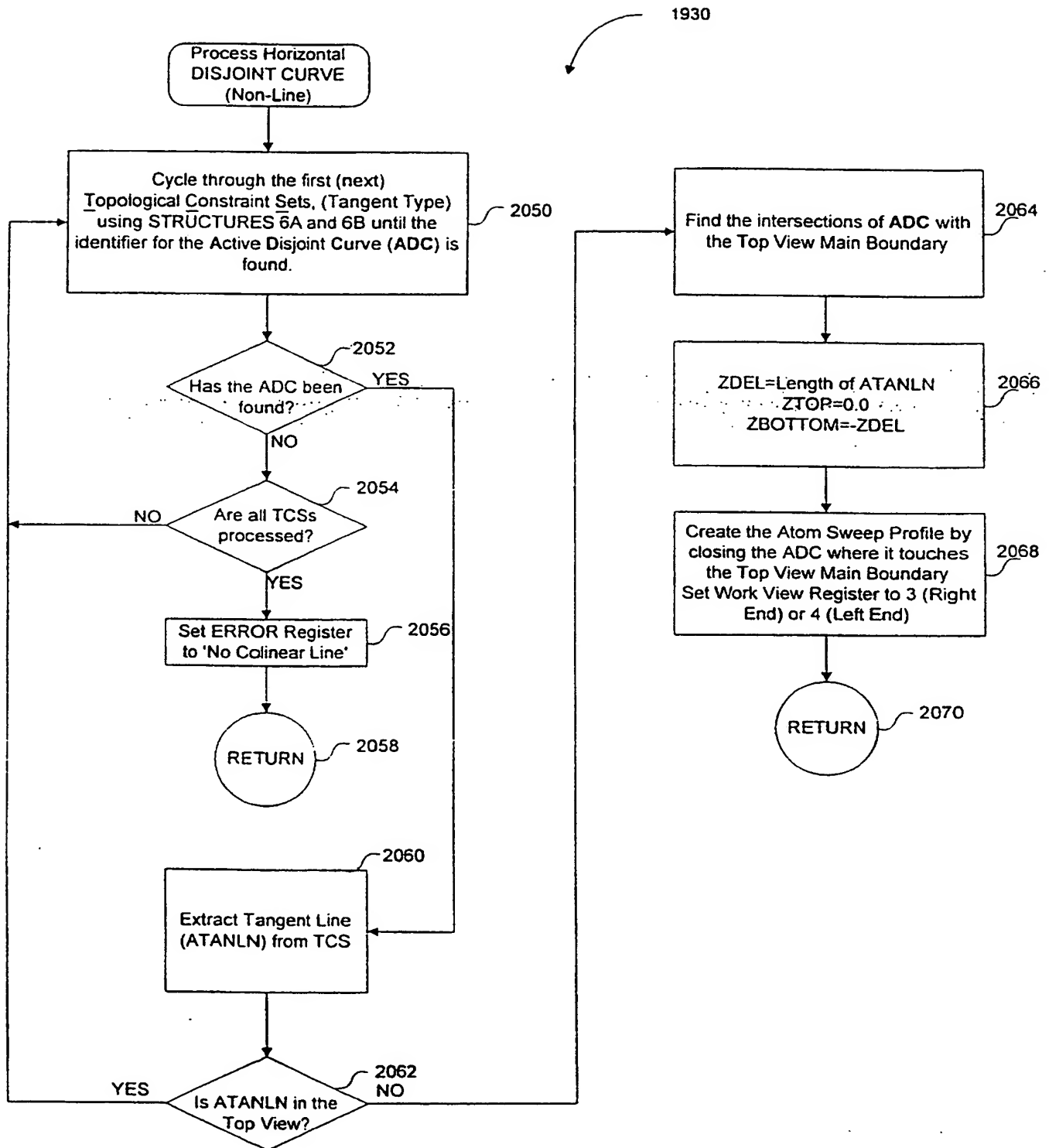


Fig. 13I

32/64

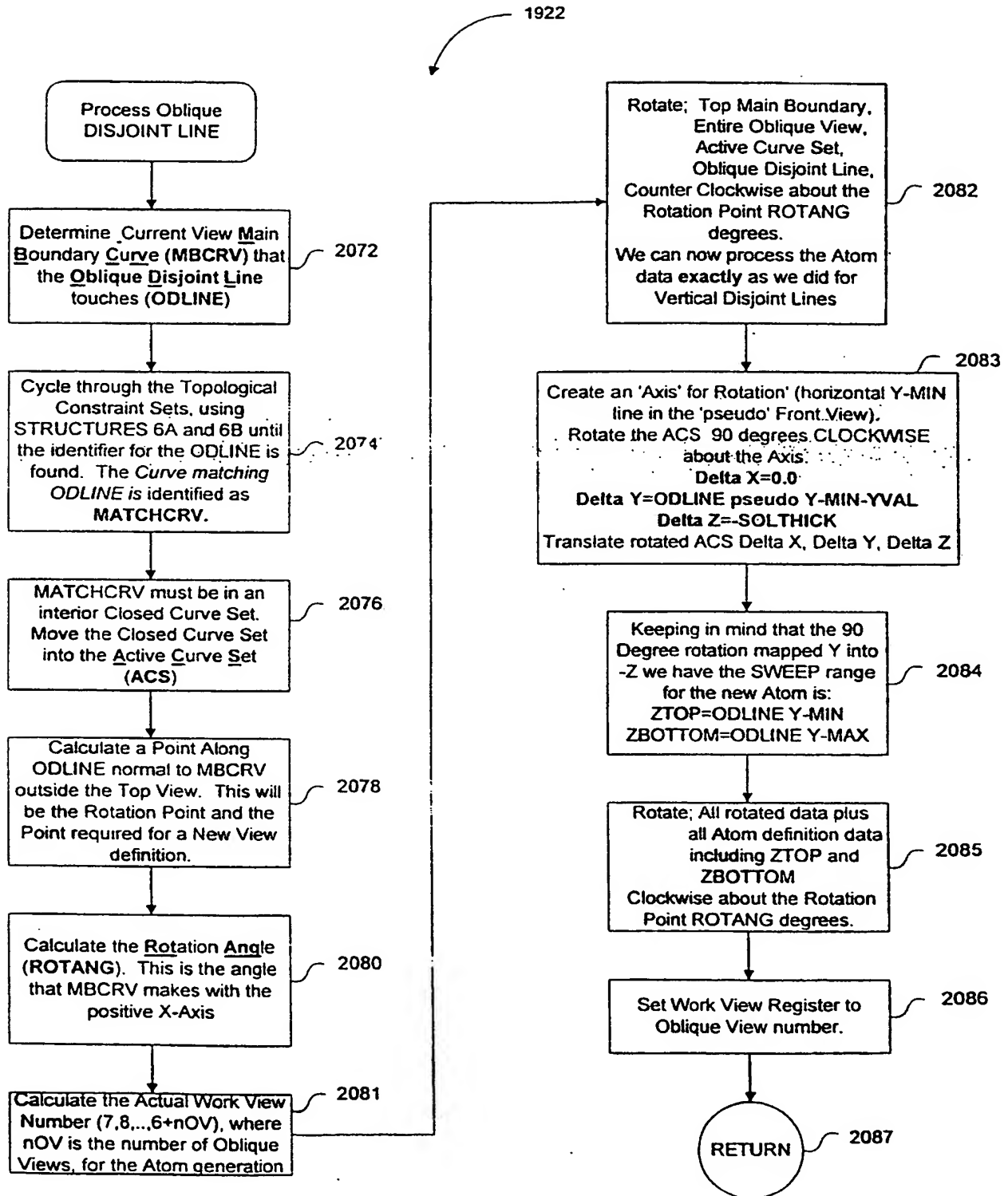
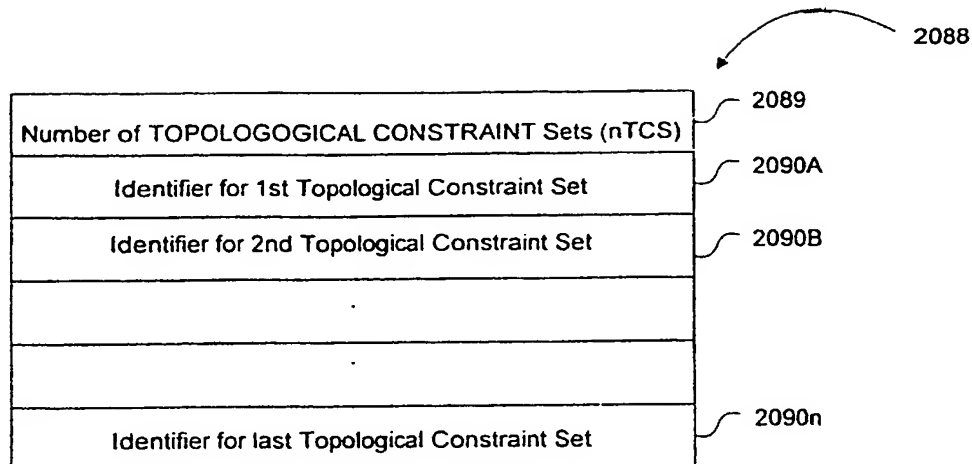
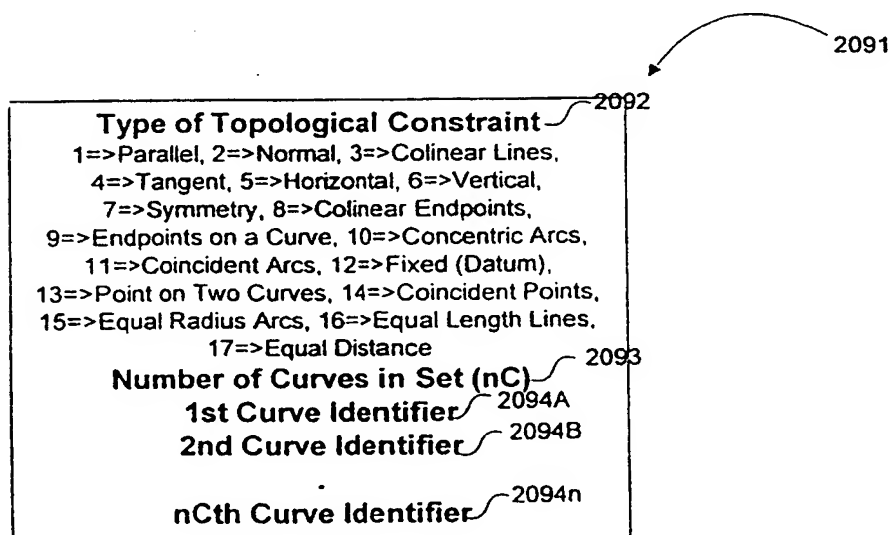


Fig. 13J

33/64



STRUCTURE 6A
(TOPOLOGICAL CONSTRAINT IDENTIFIERS)



STRUCTURE 6B
(TOPOLOGICAL CONSTRAINT SET)

Fig. 13K

34/64

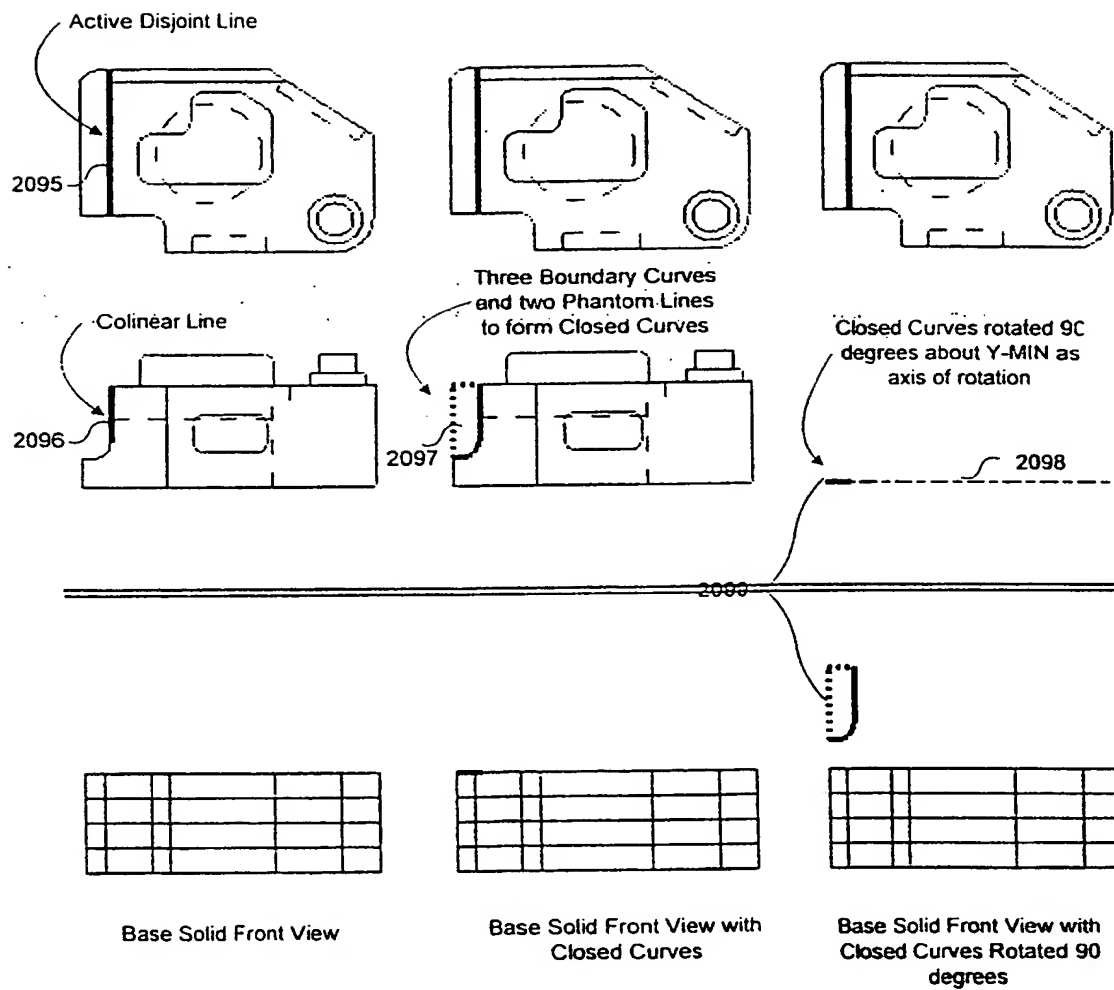


Fig. 13L

35/64

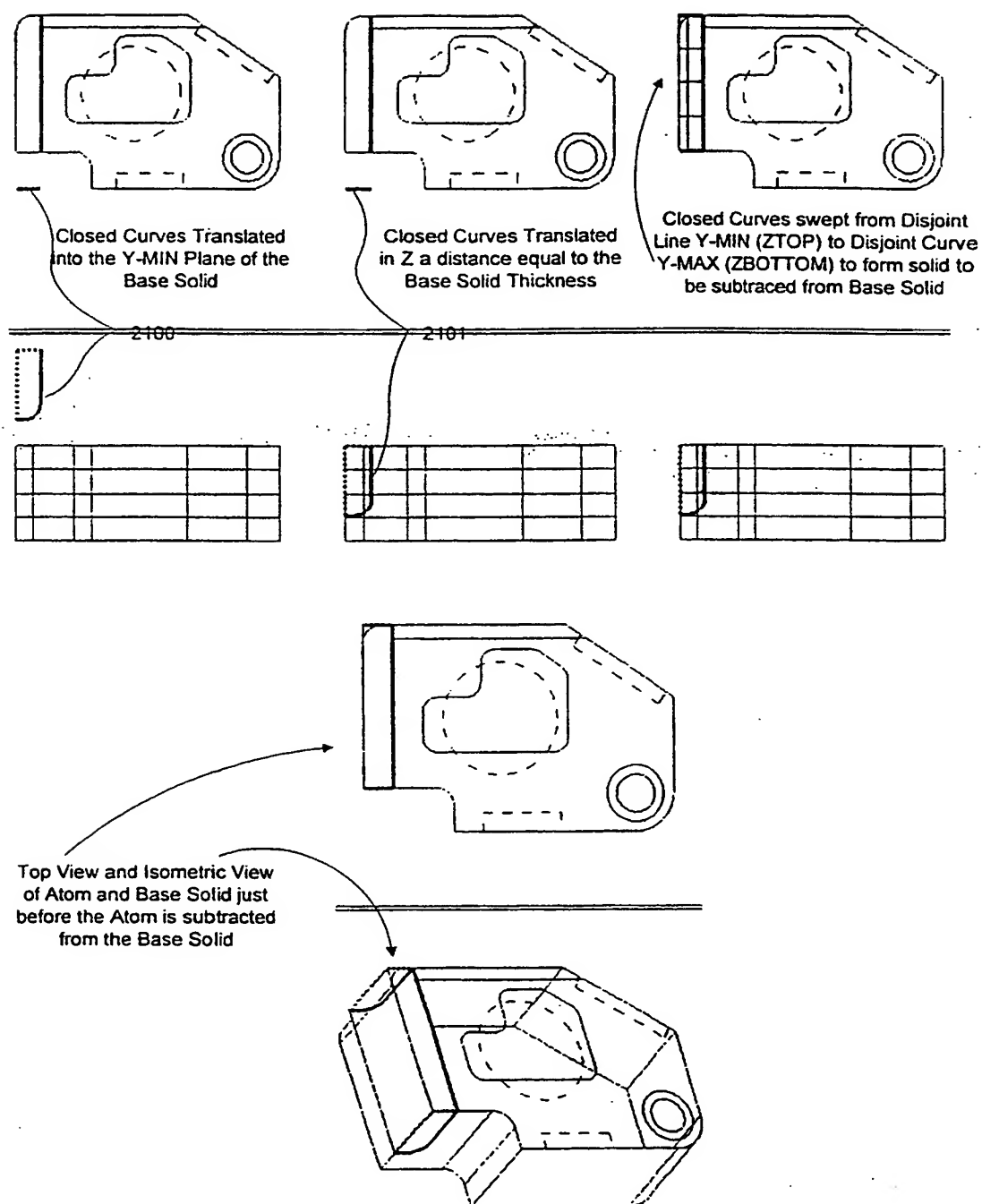


Fig. 13M

36/64

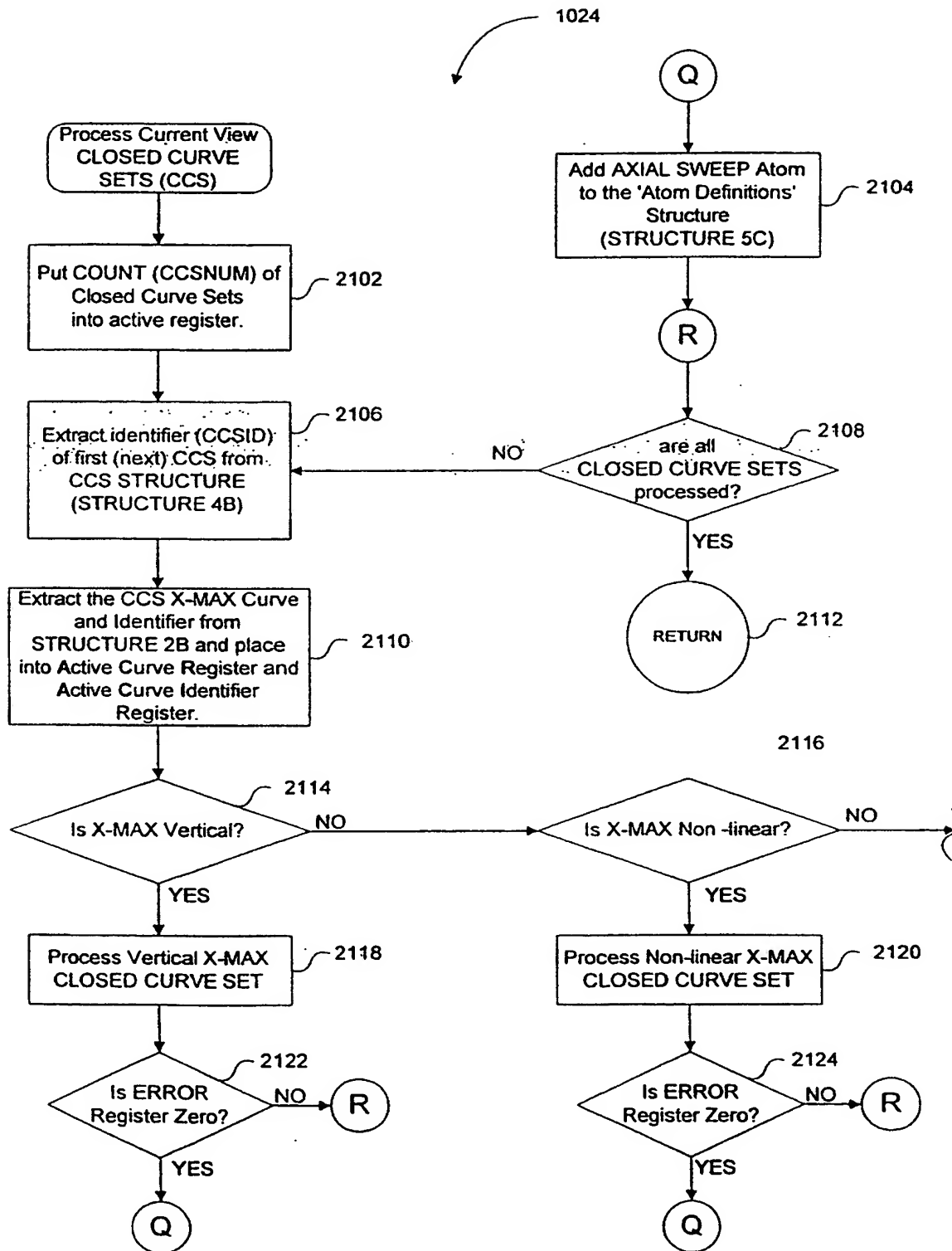


Fig. 14A

37/64

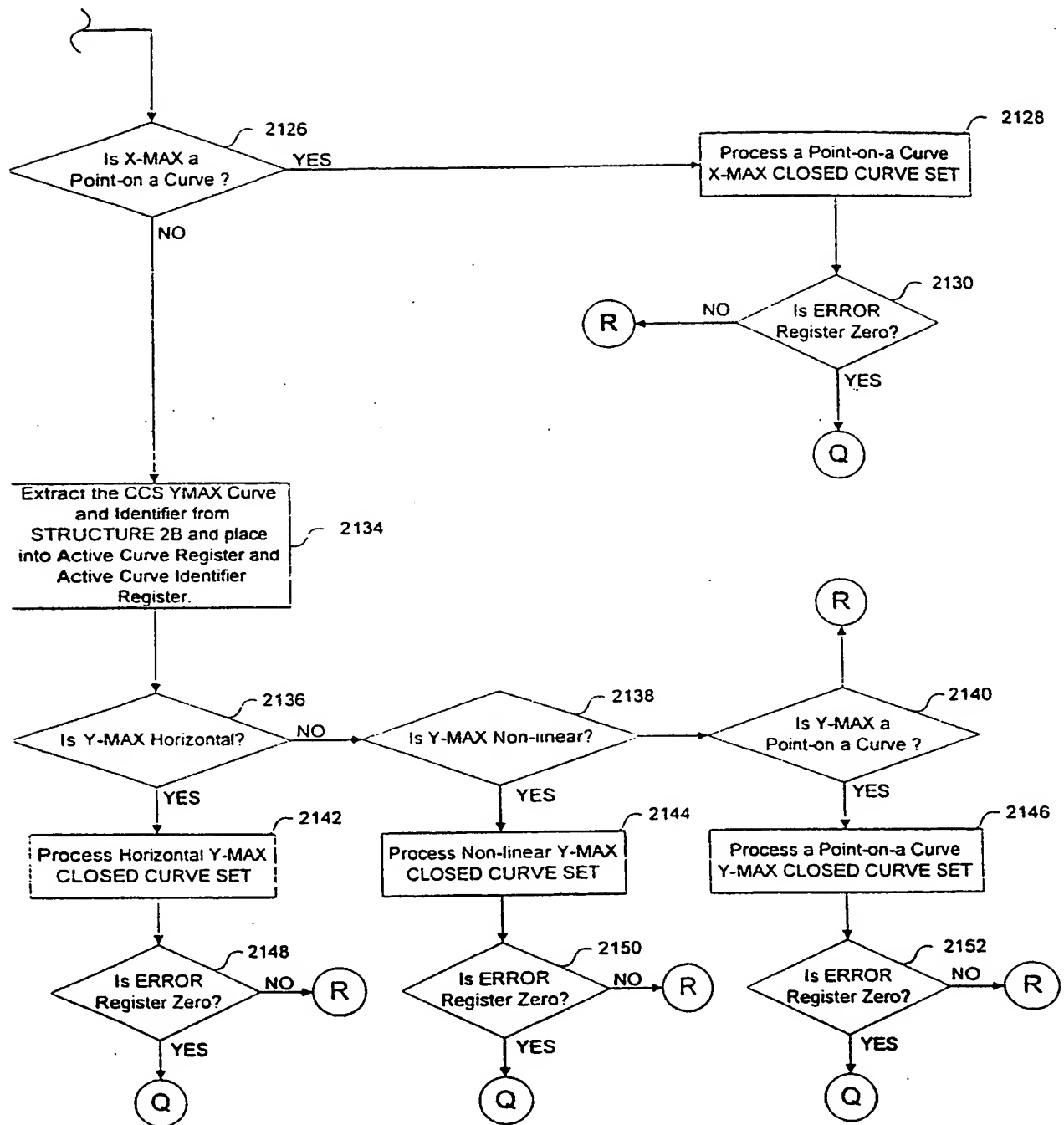


Fig. 14B

38/64

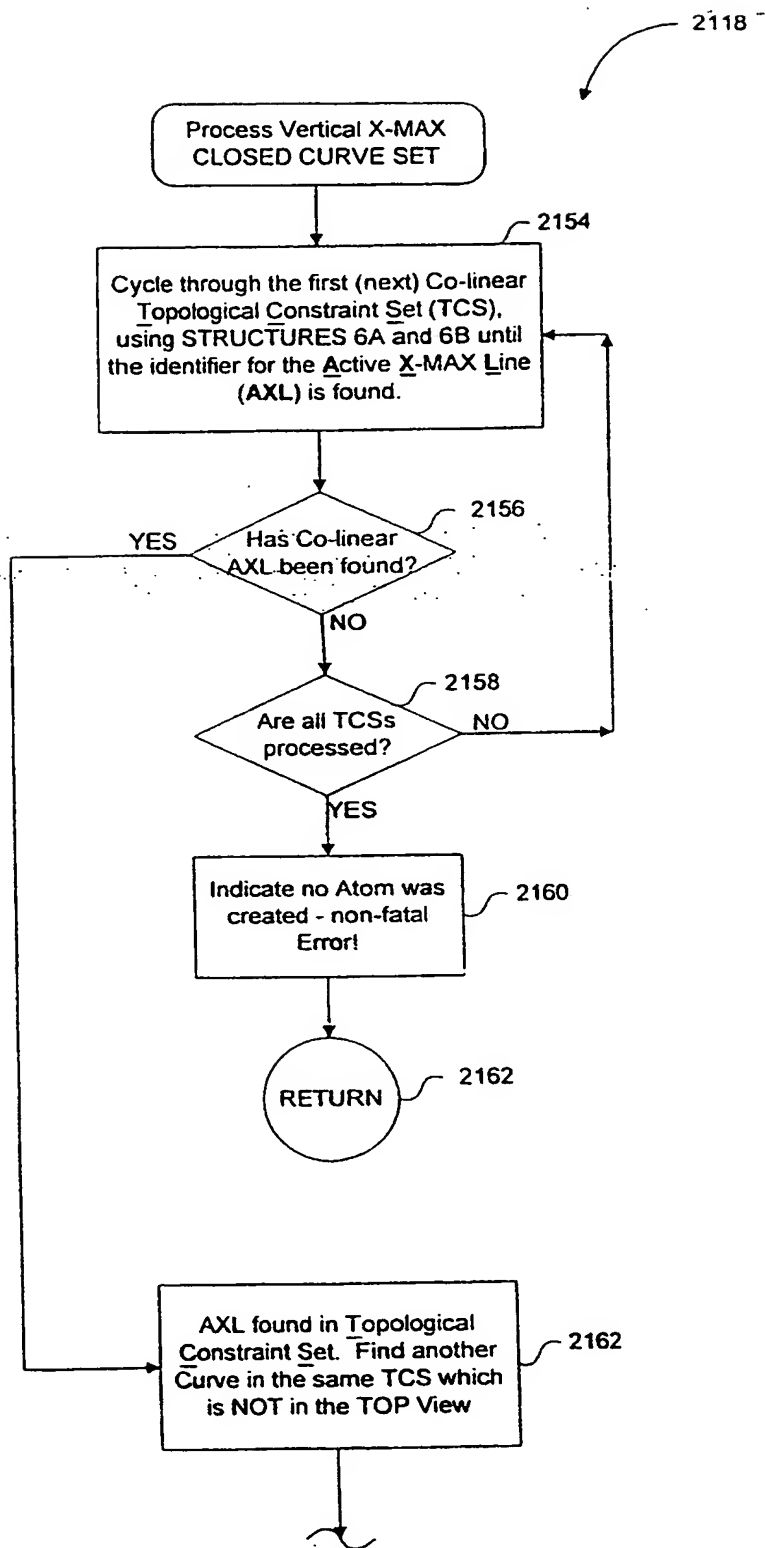


Fig. 14C

39/64

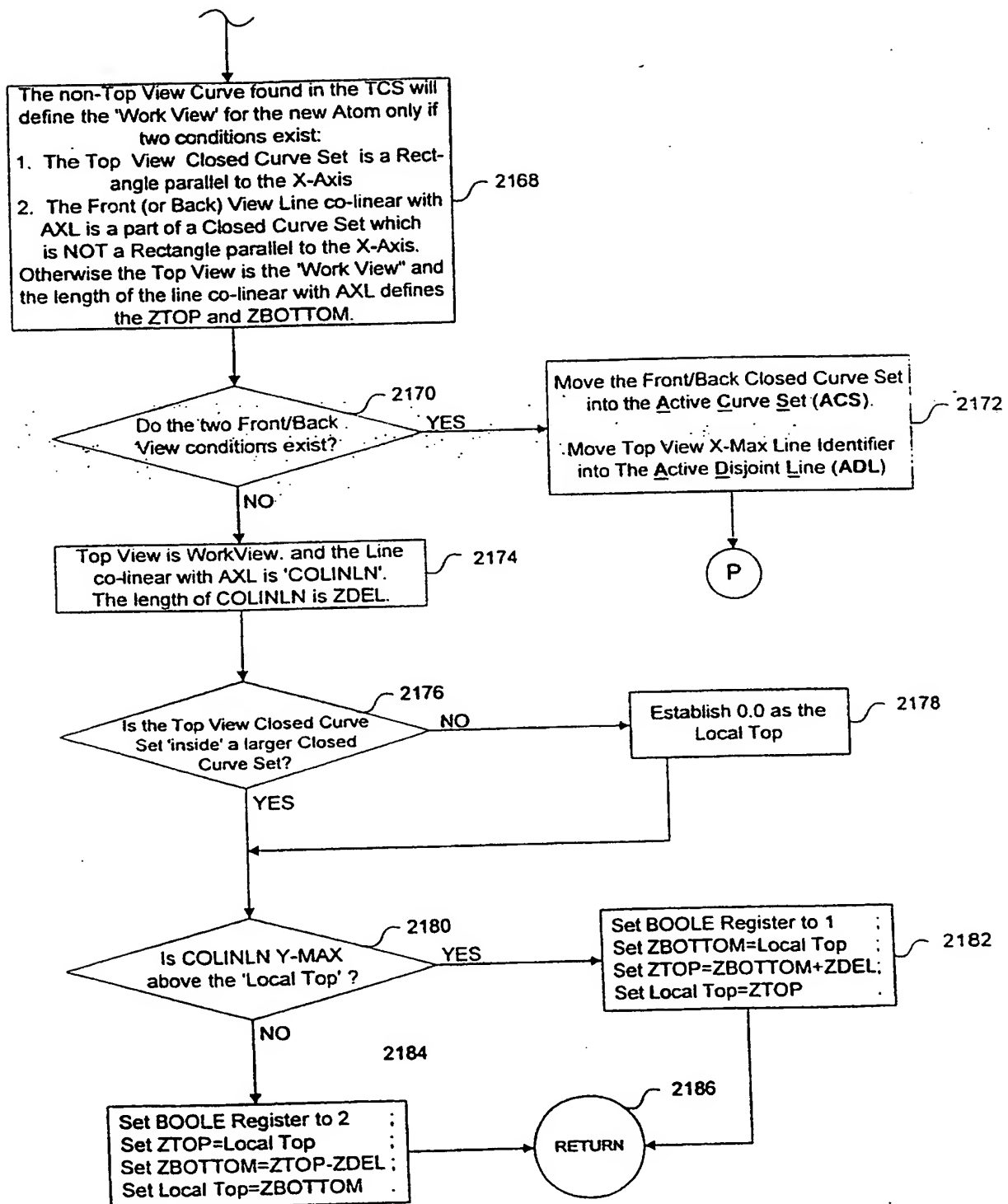


Fig. 14D

40/64

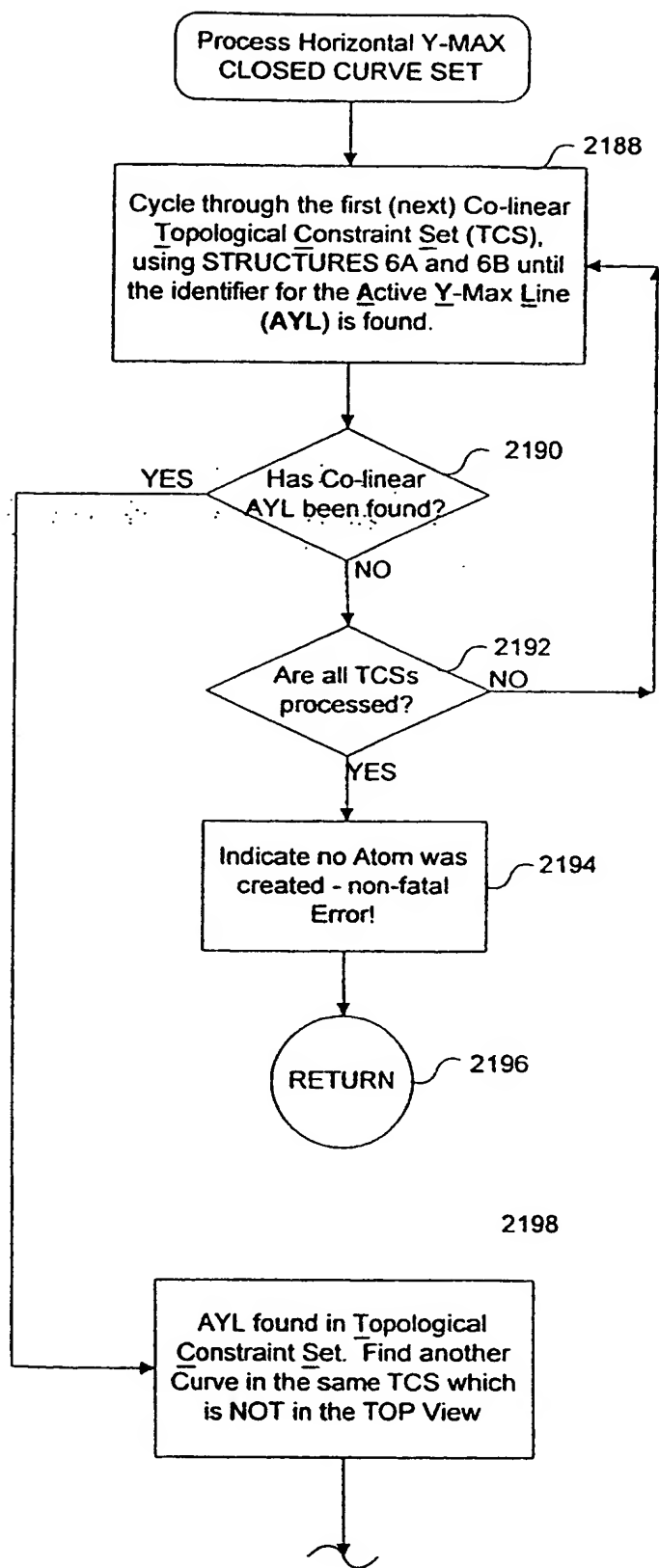


Fig. 14E

41/64

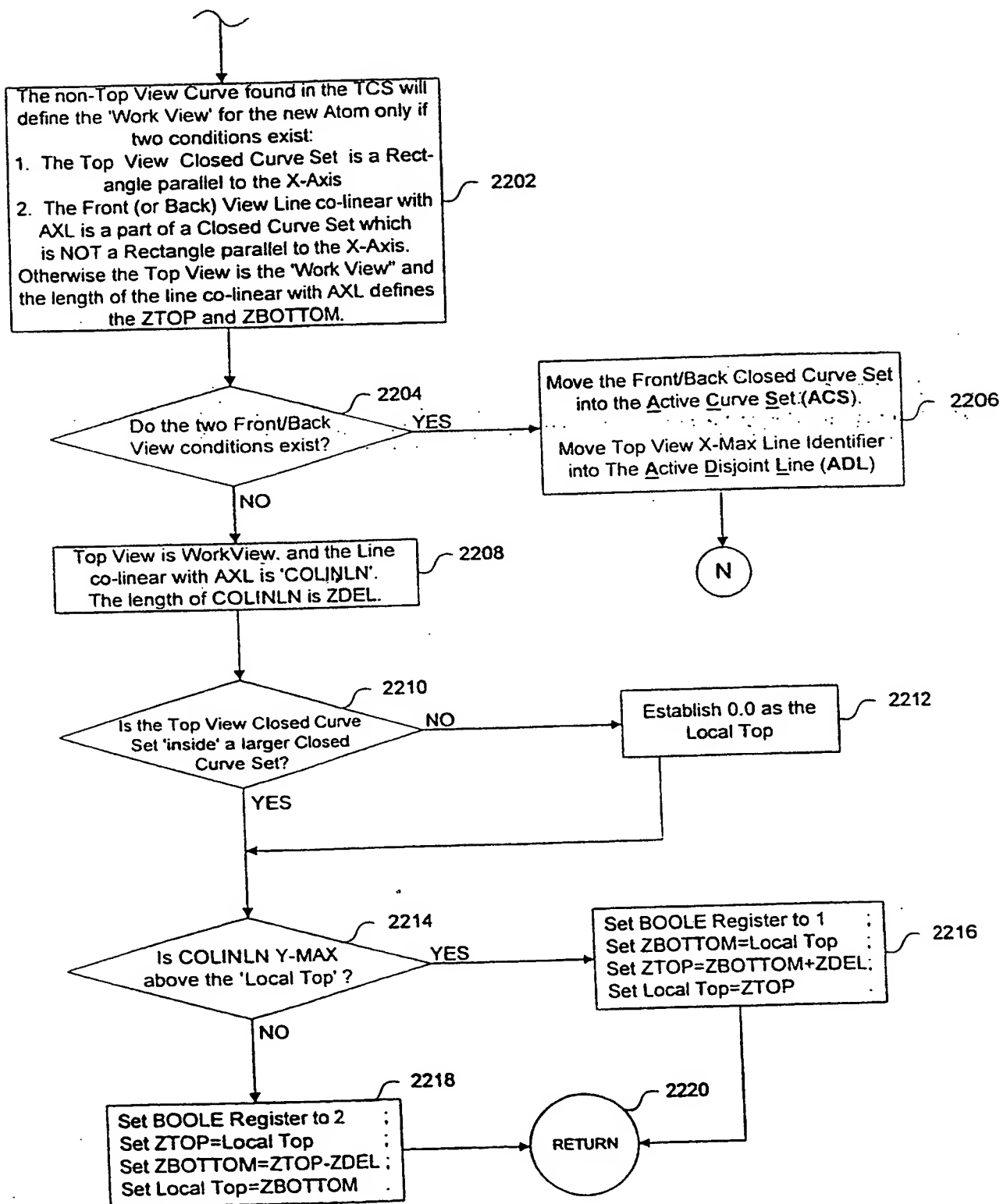


Fig. 14F

42/64

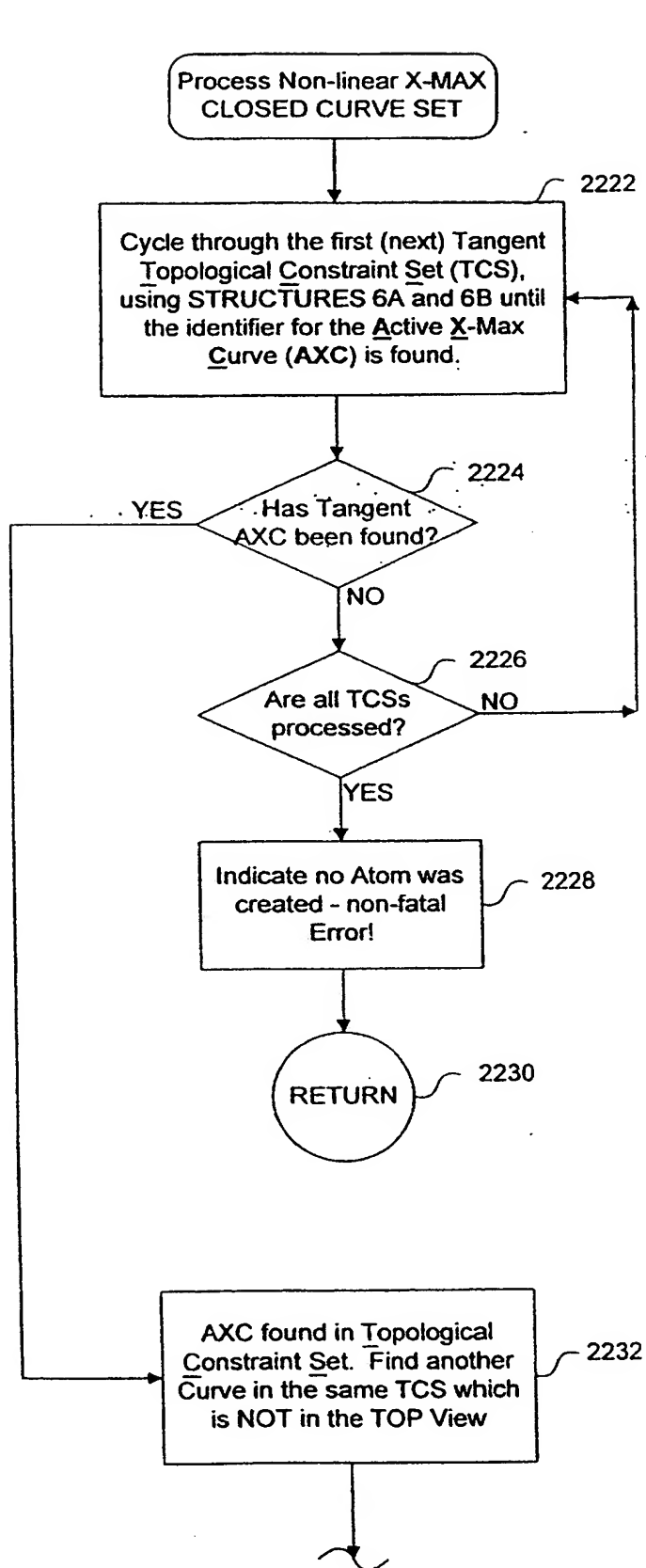


Fig. 14G

43/64

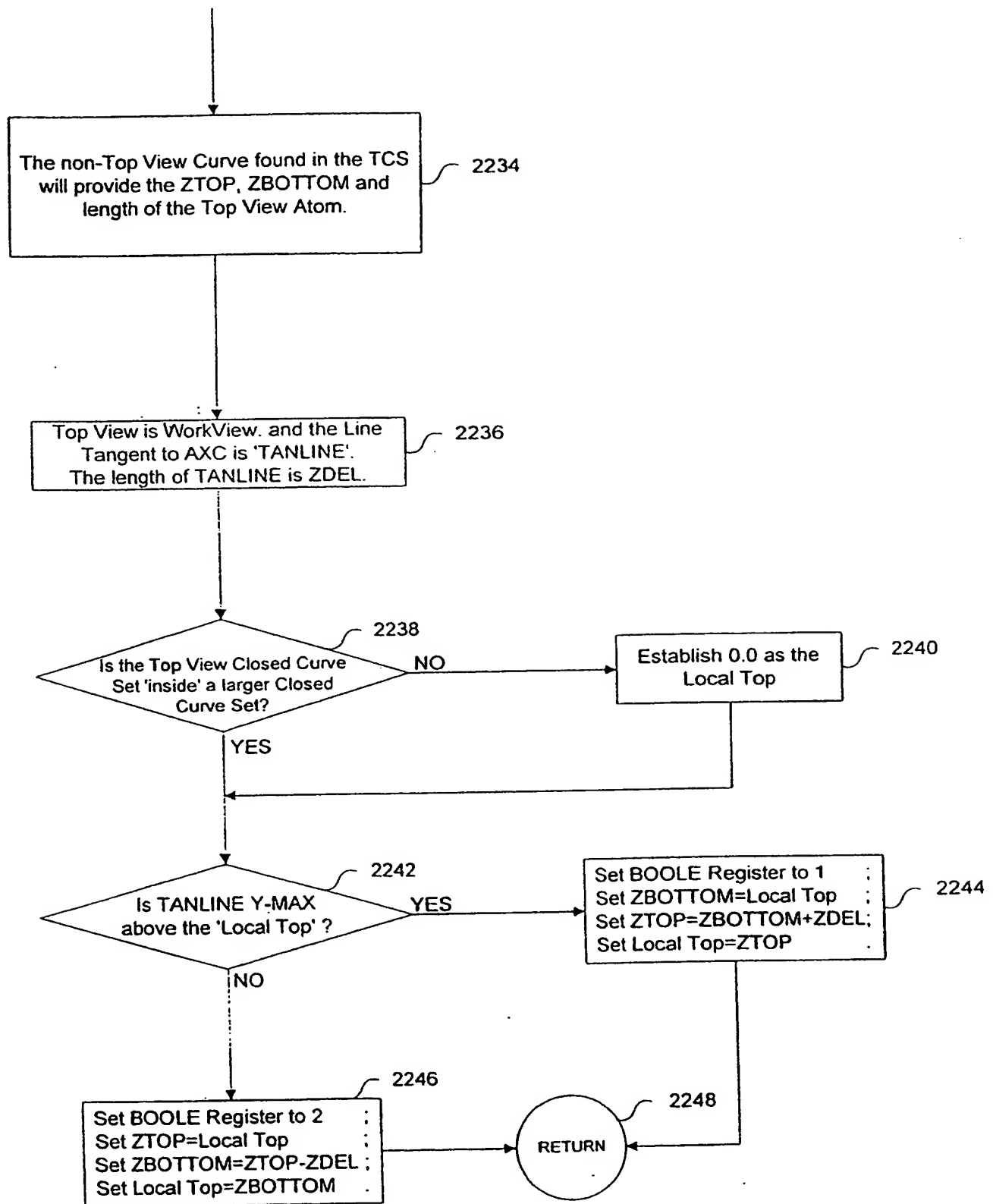


Fig. 14H

44/64

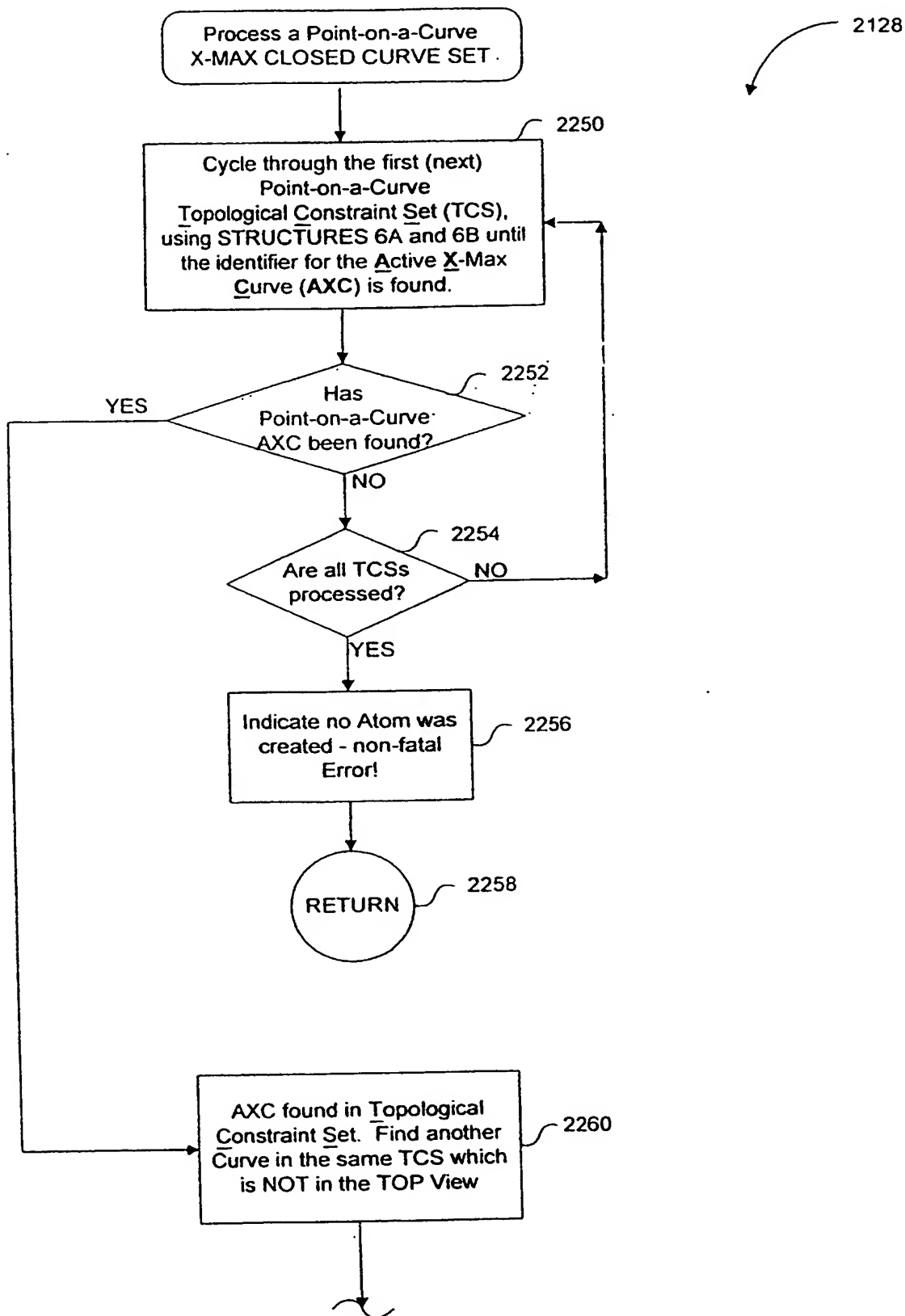


Fig. 14I

45/64

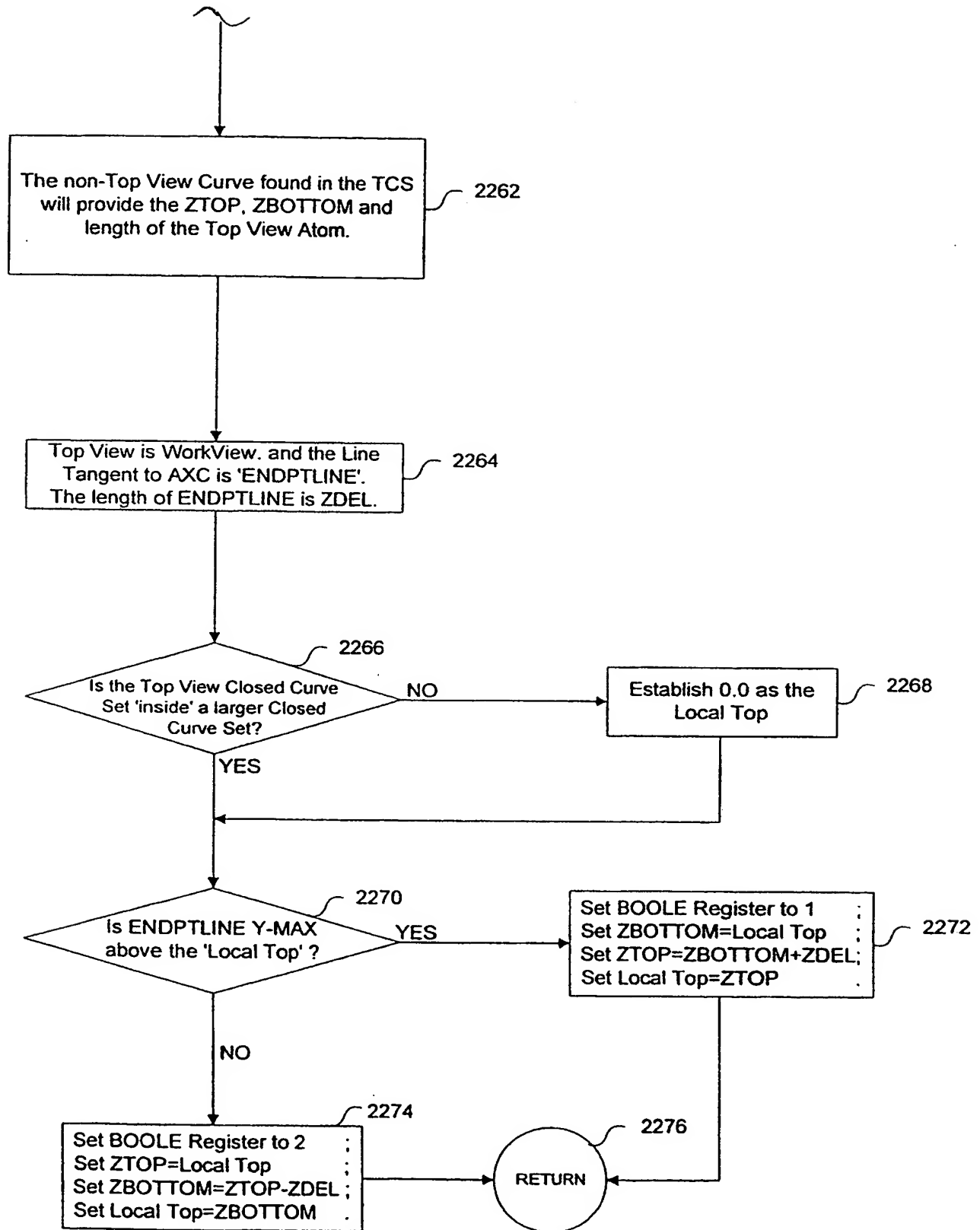


Fig. 14J

46/64

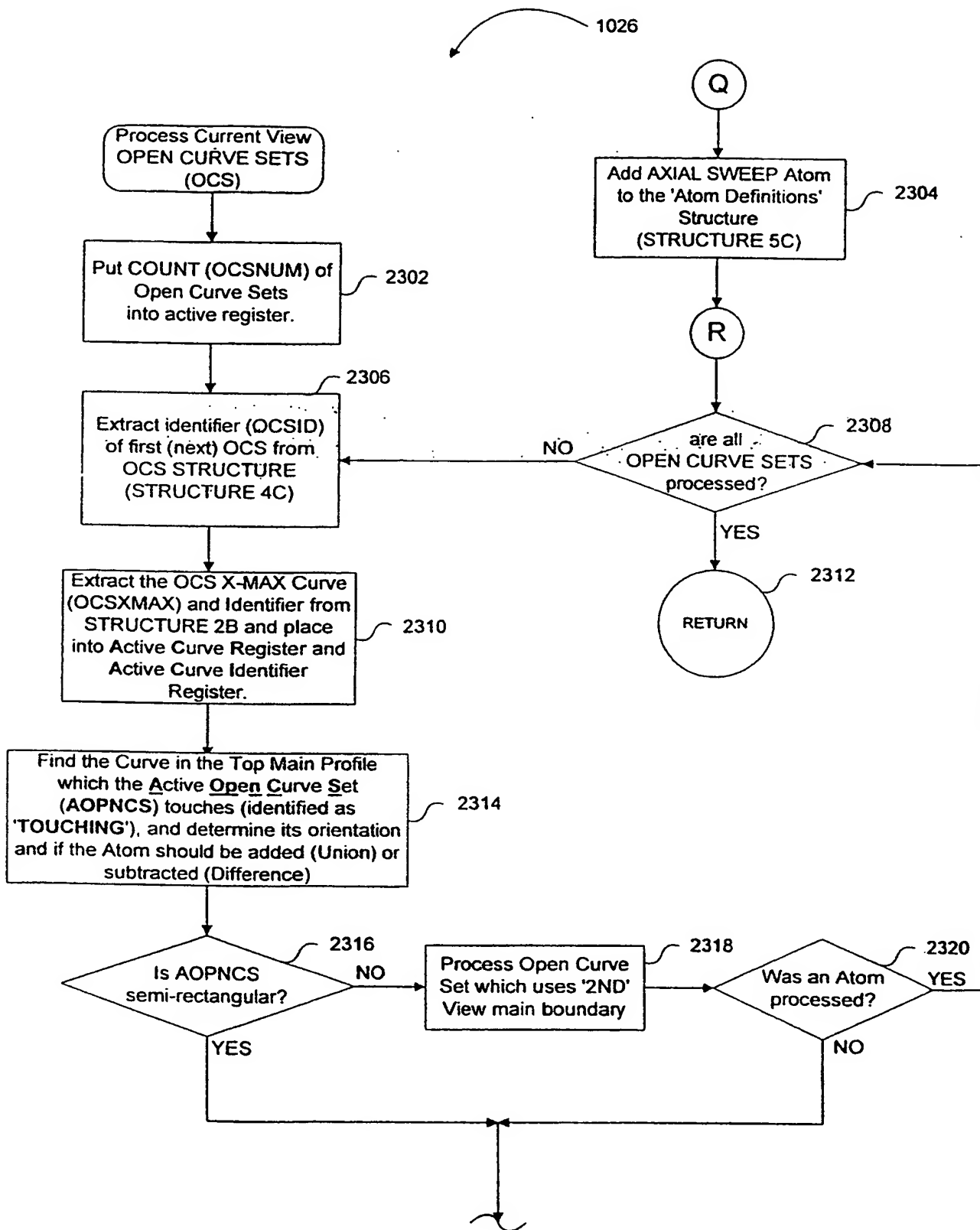


Fig. 15A

47/64

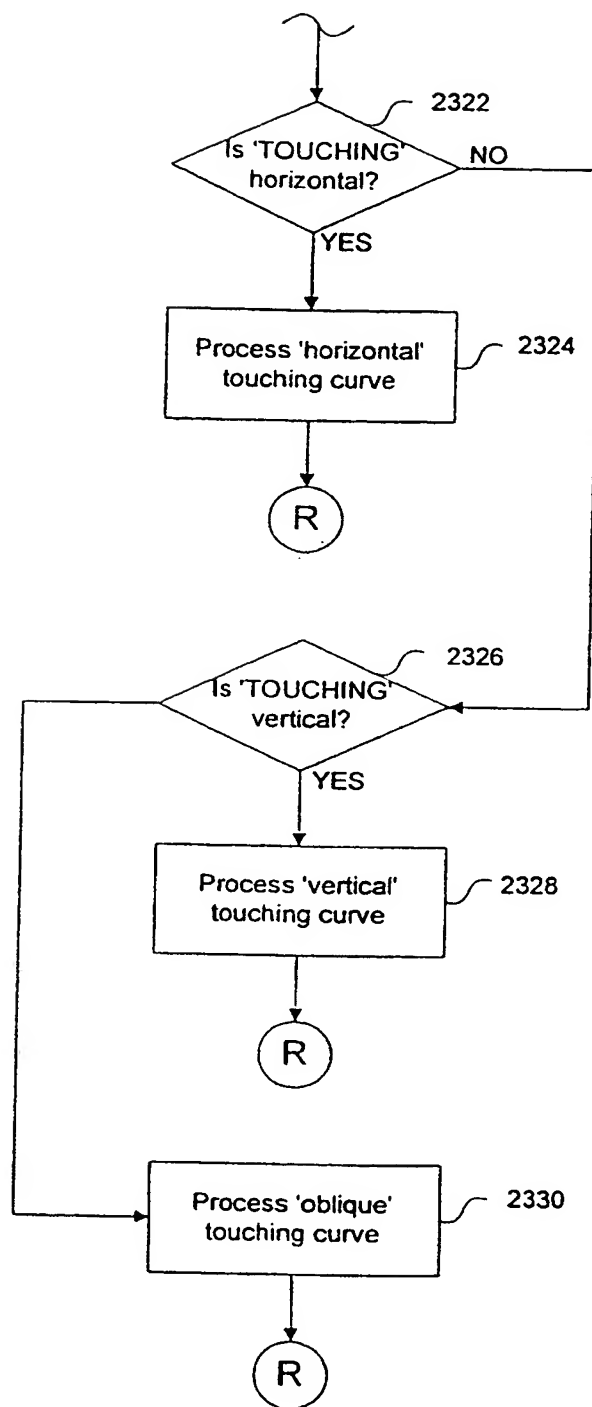


Fig. 15B

48/64

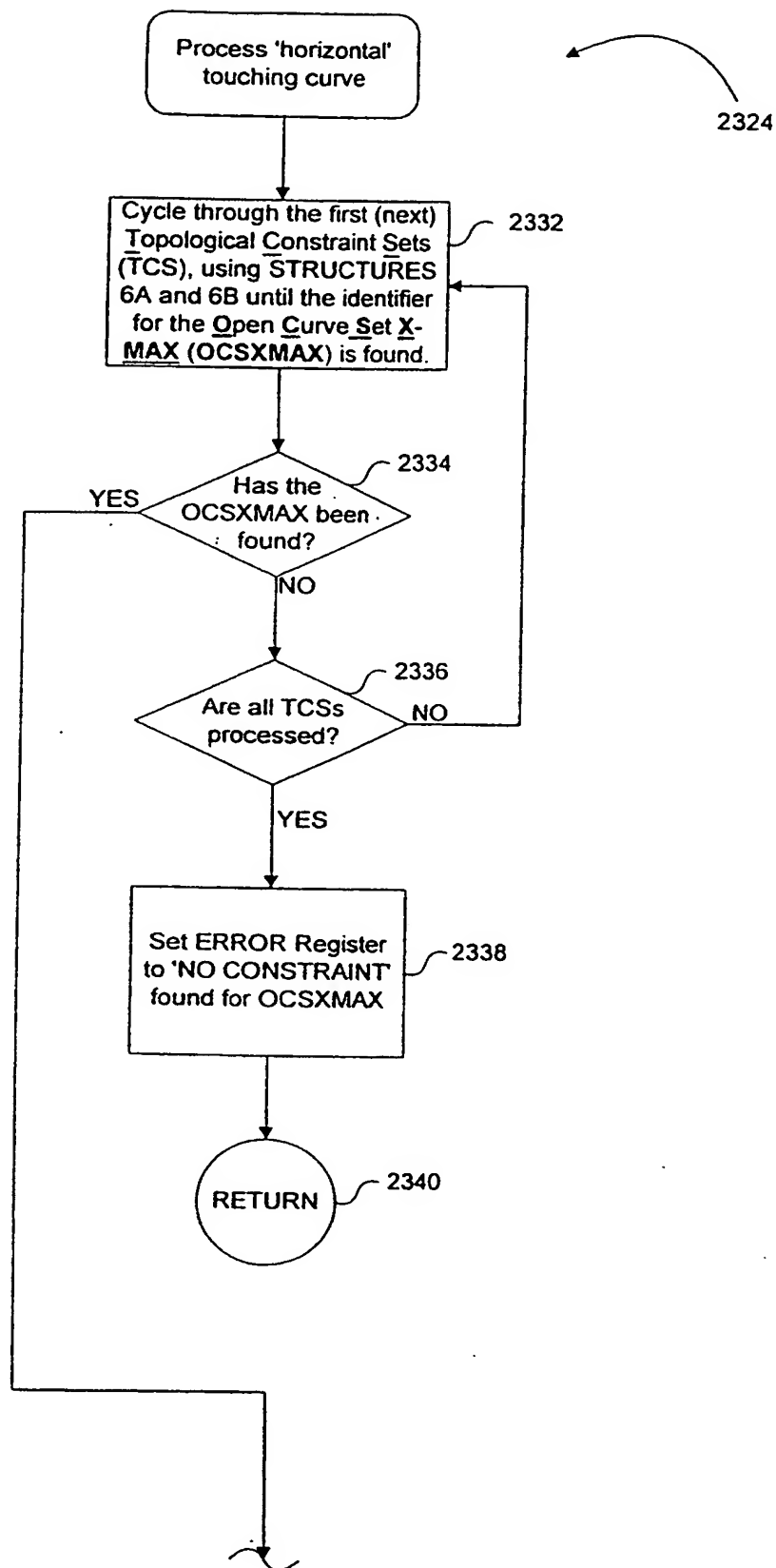


Fig. 15C

49/64

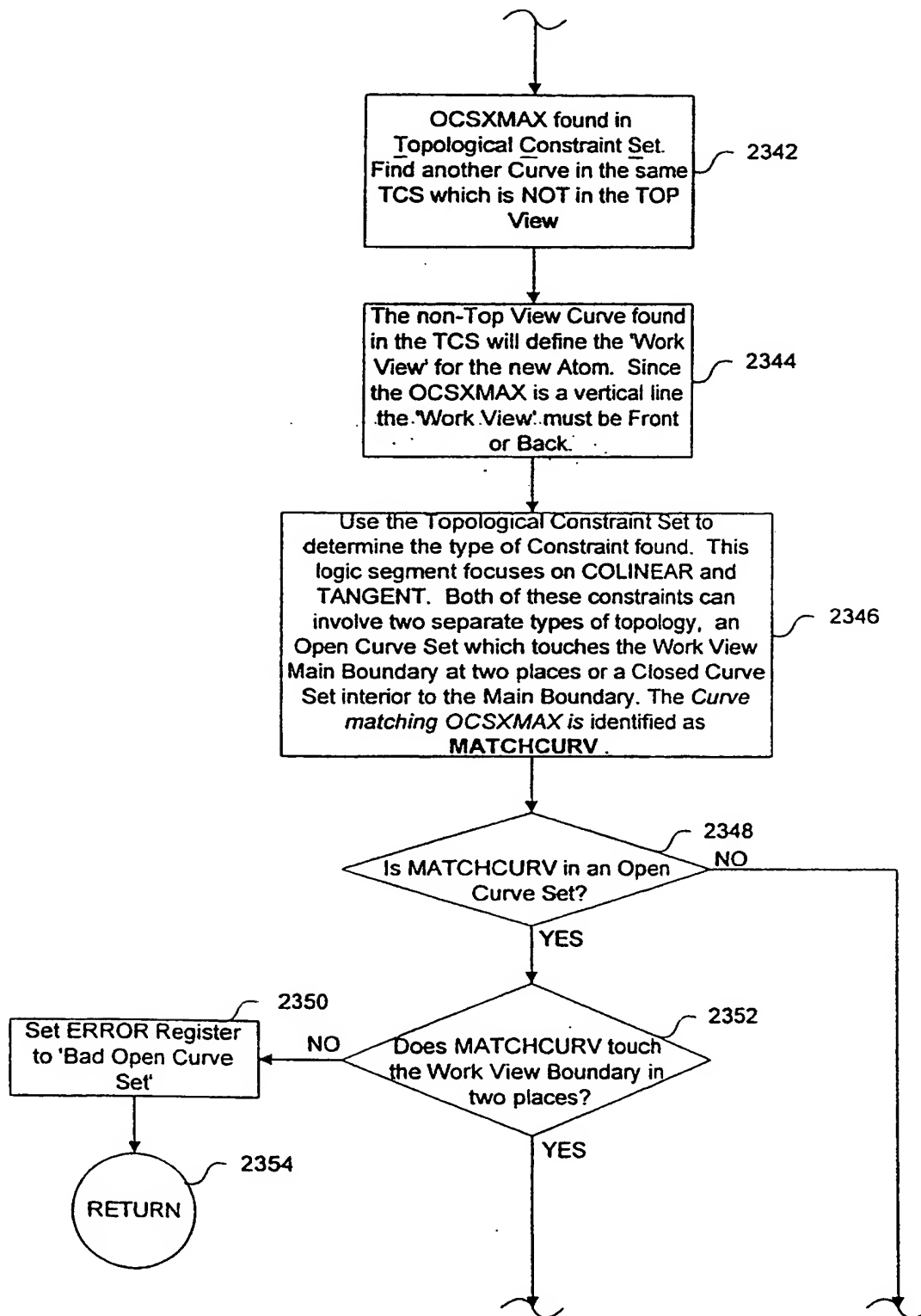


Fig. 15D

50/64

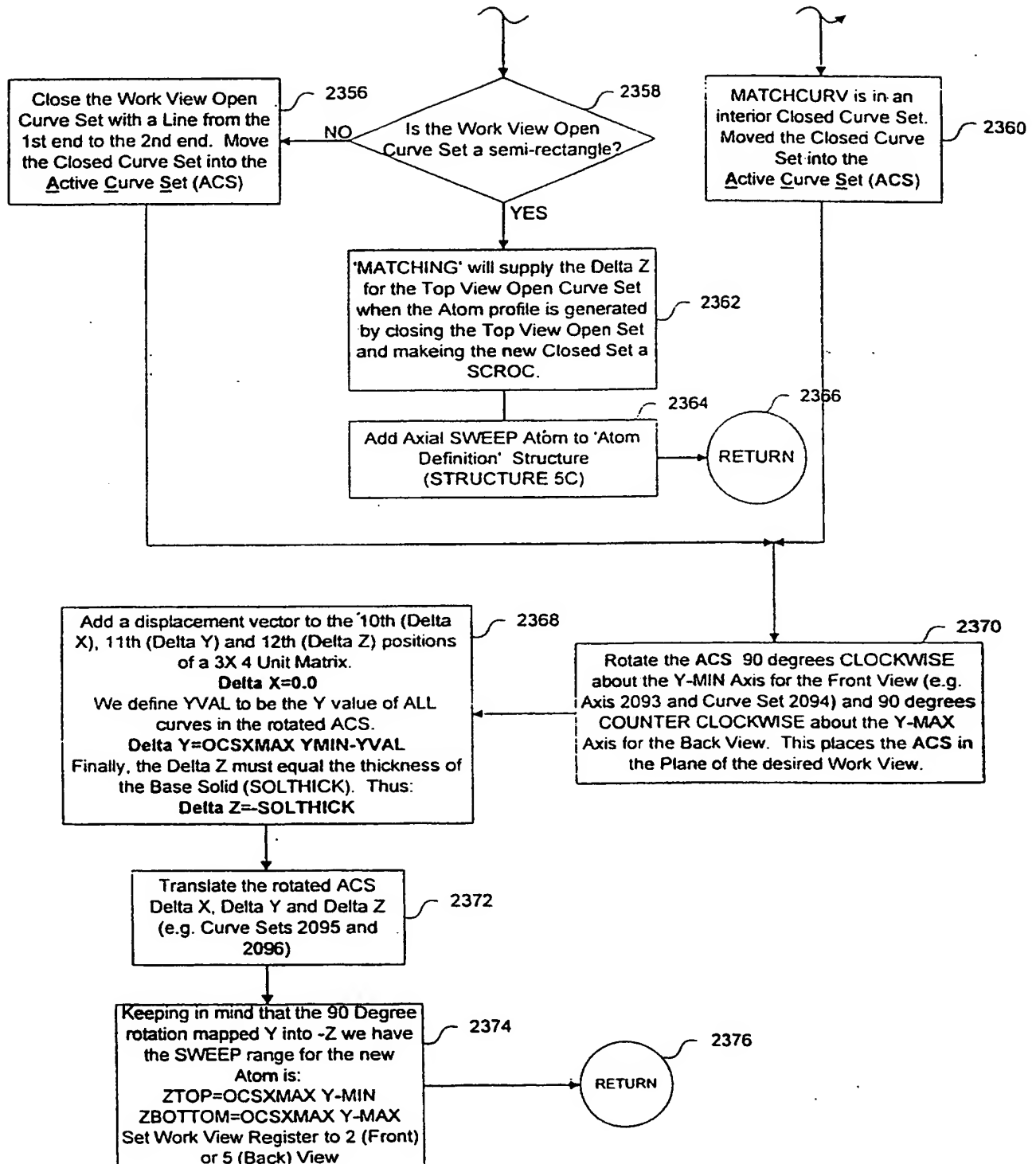


Fig. 15E

51/64

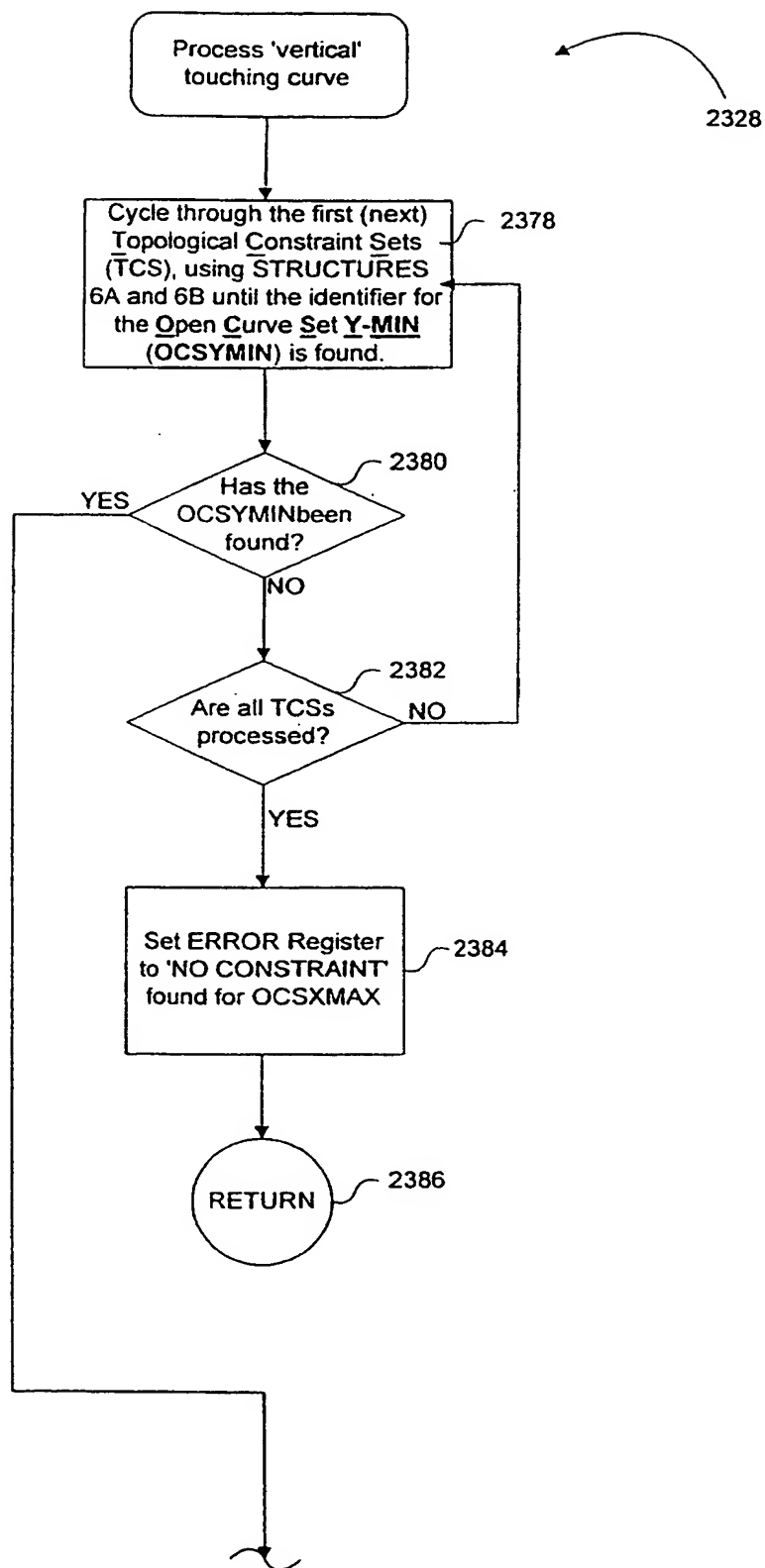


Fig. 15F

52/64

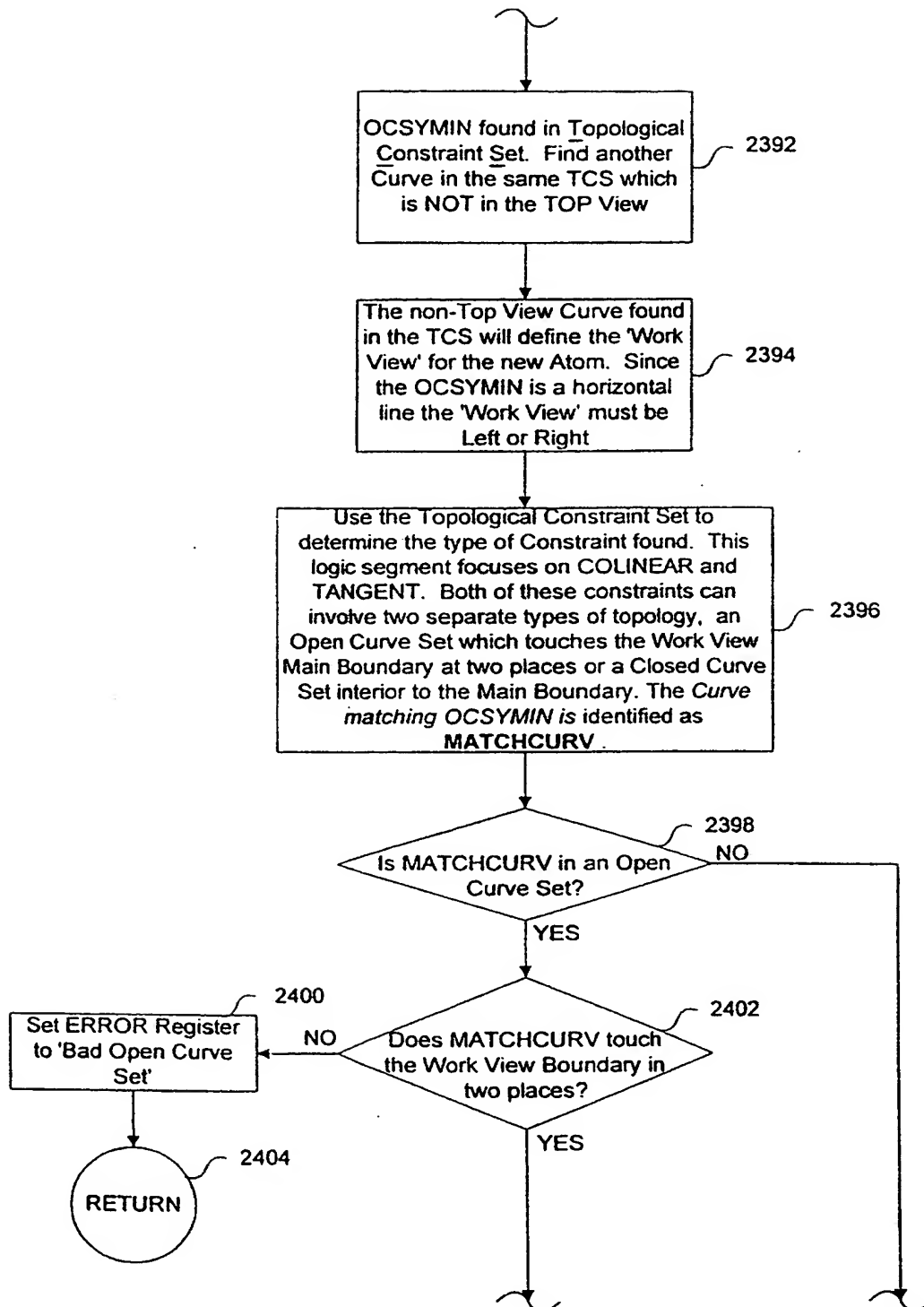


Fig. 15G

53/64

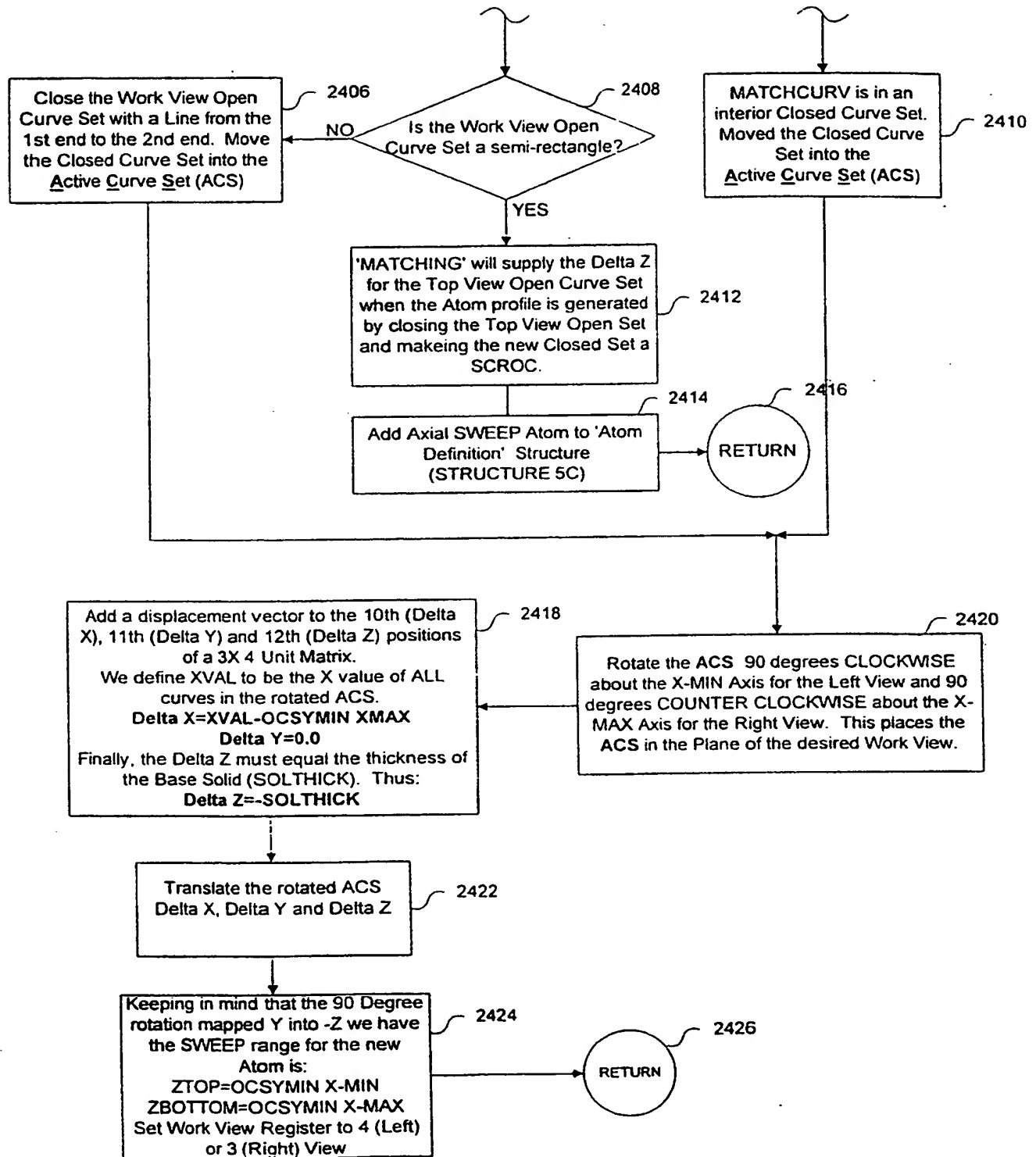


Fig. 15H

54/64

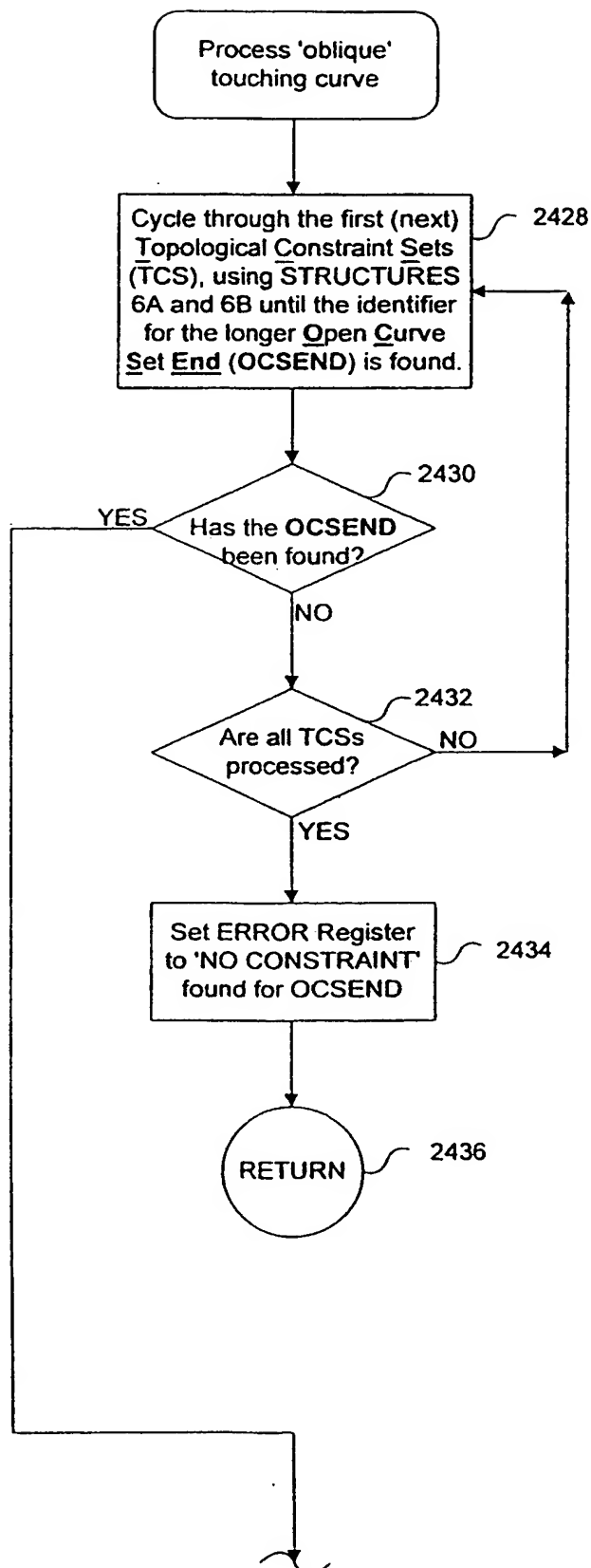


Fig. 15I

55/64

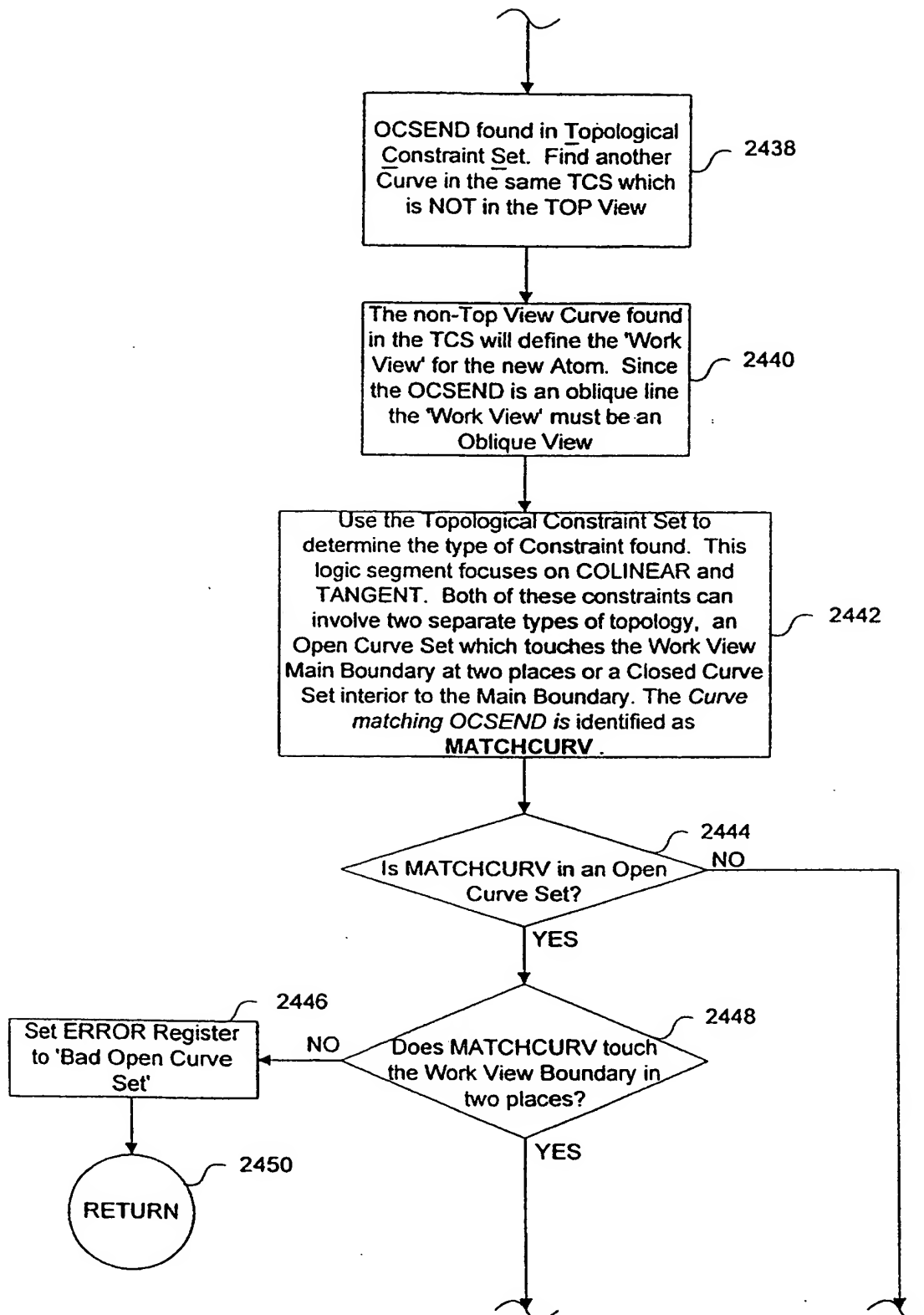


Fig. 15J

56/64

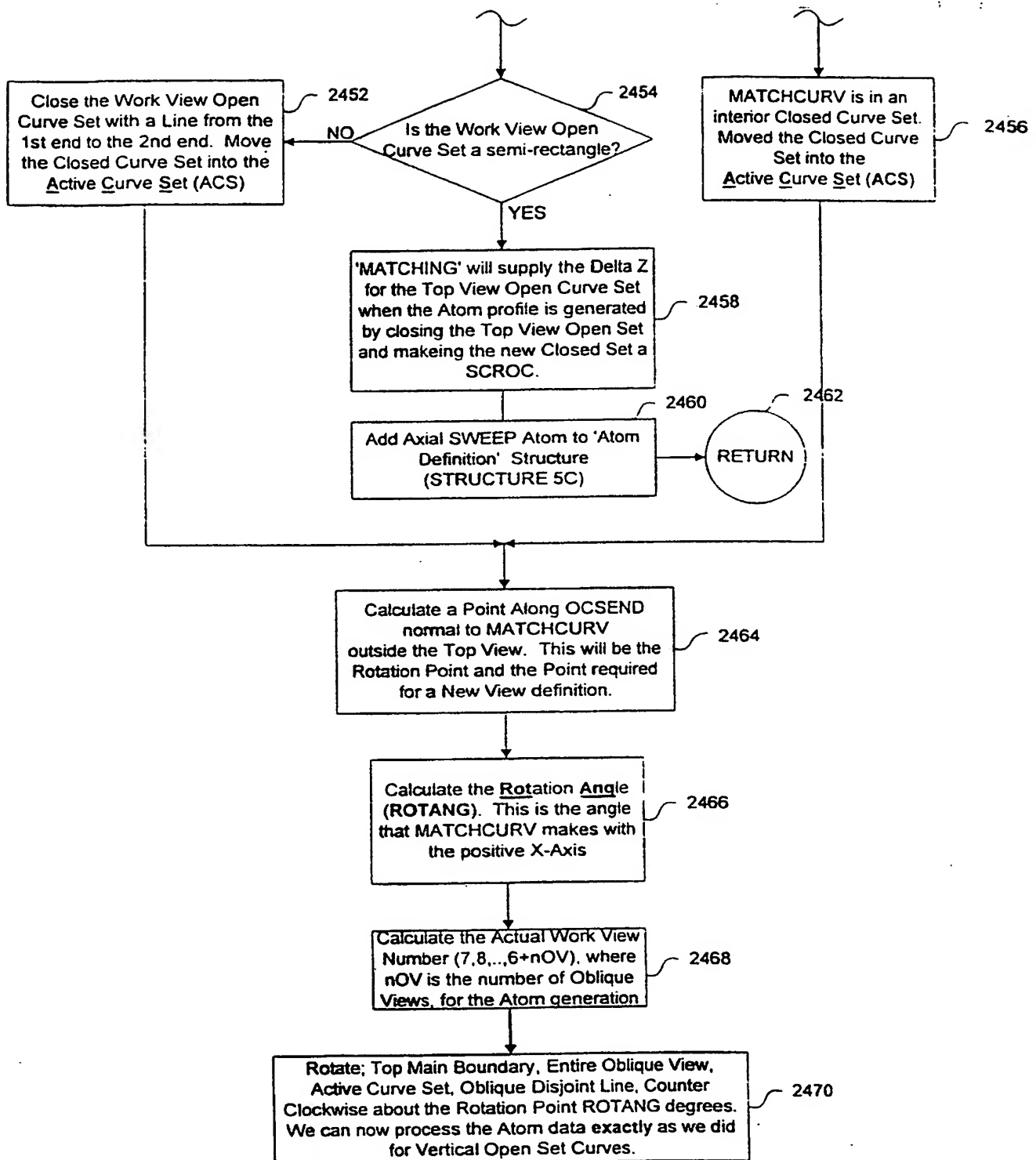


Fig. 15K

57/64

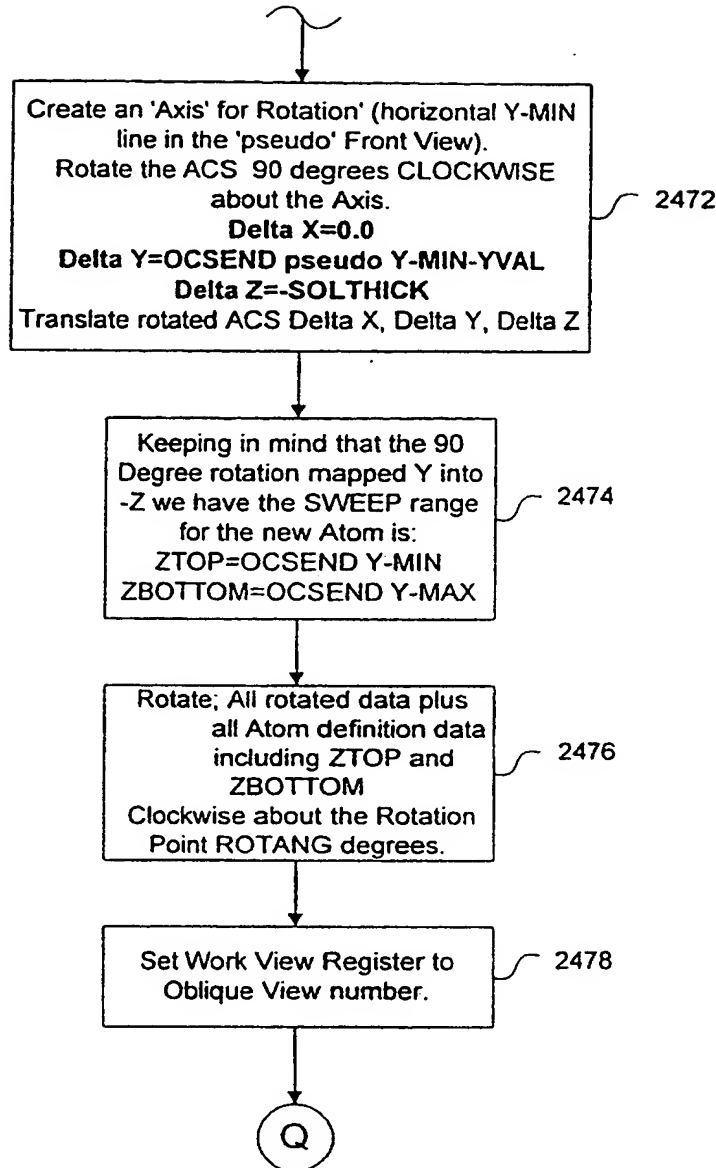


Fig. 15L

58/64

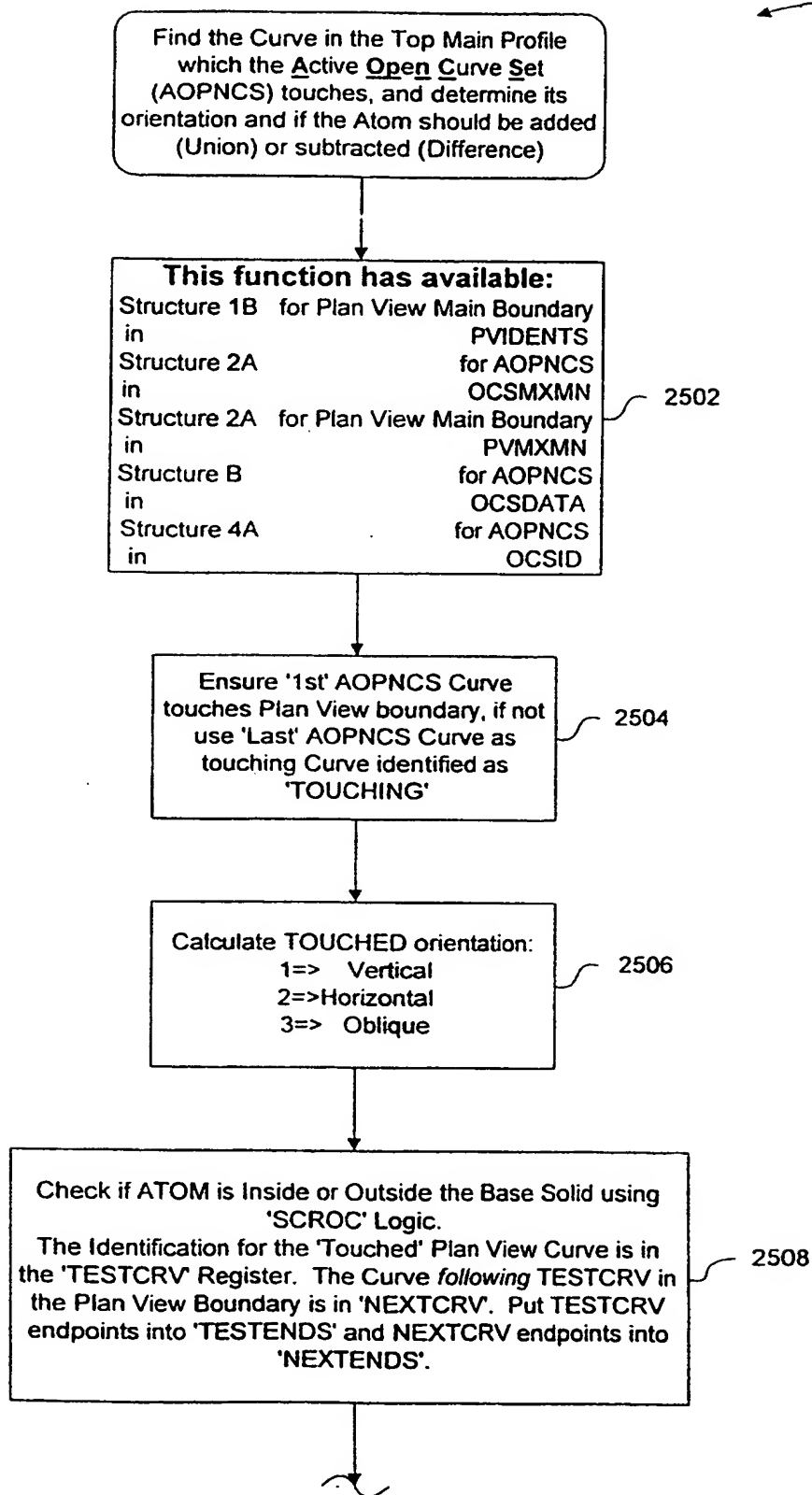


Fig. 15M

59/64

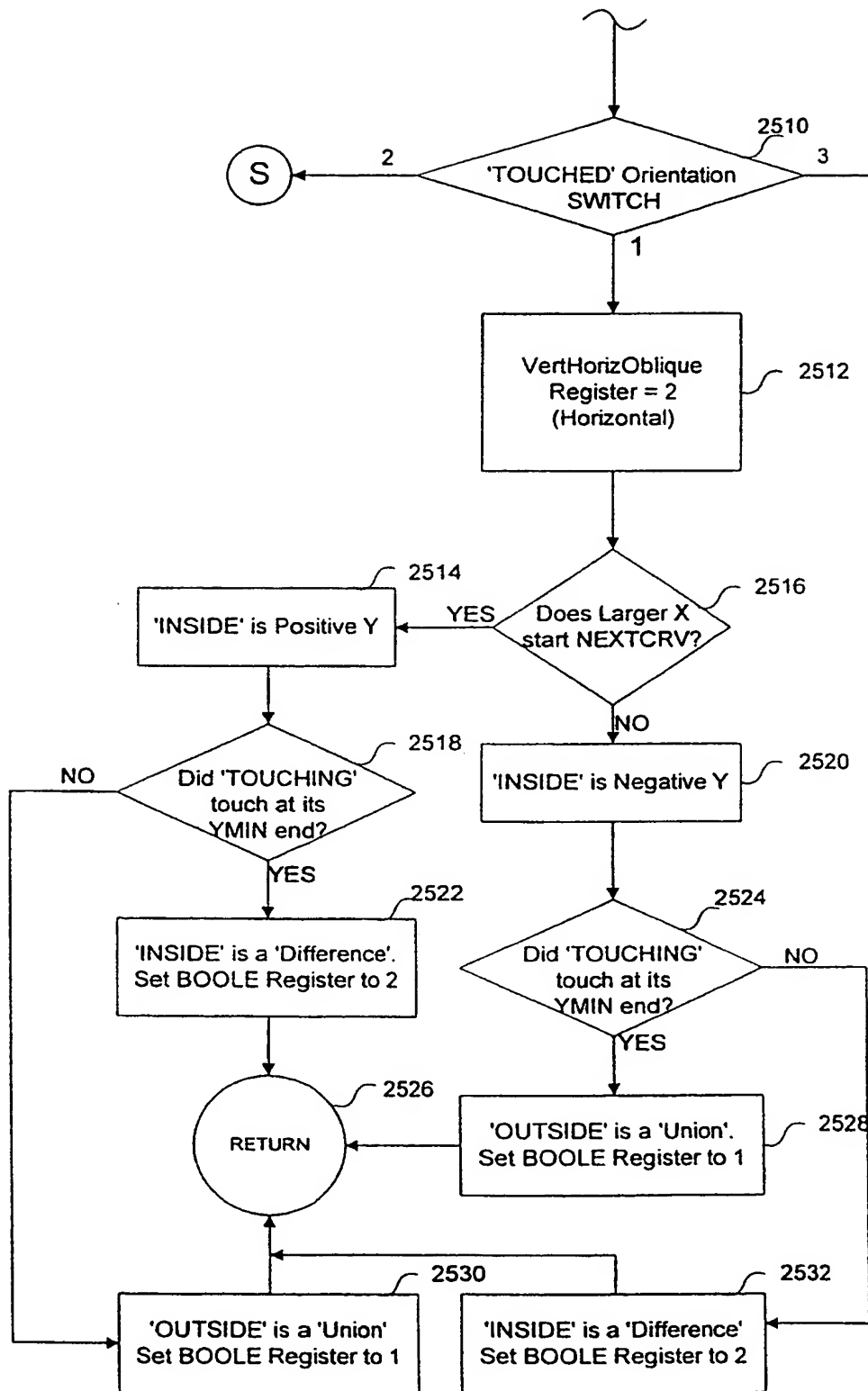


Fig. 15N

60/64

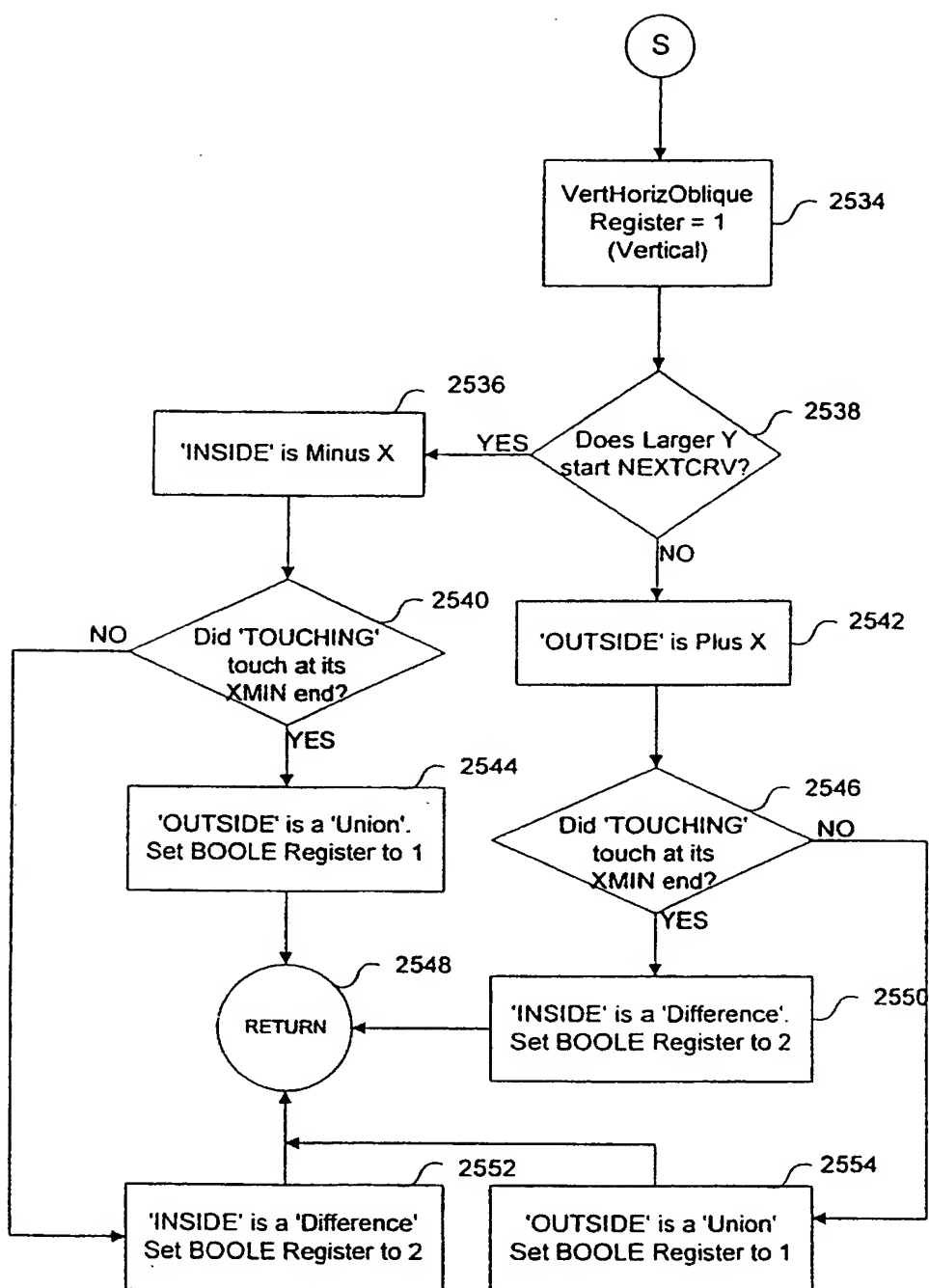


Fig. 150

61/64

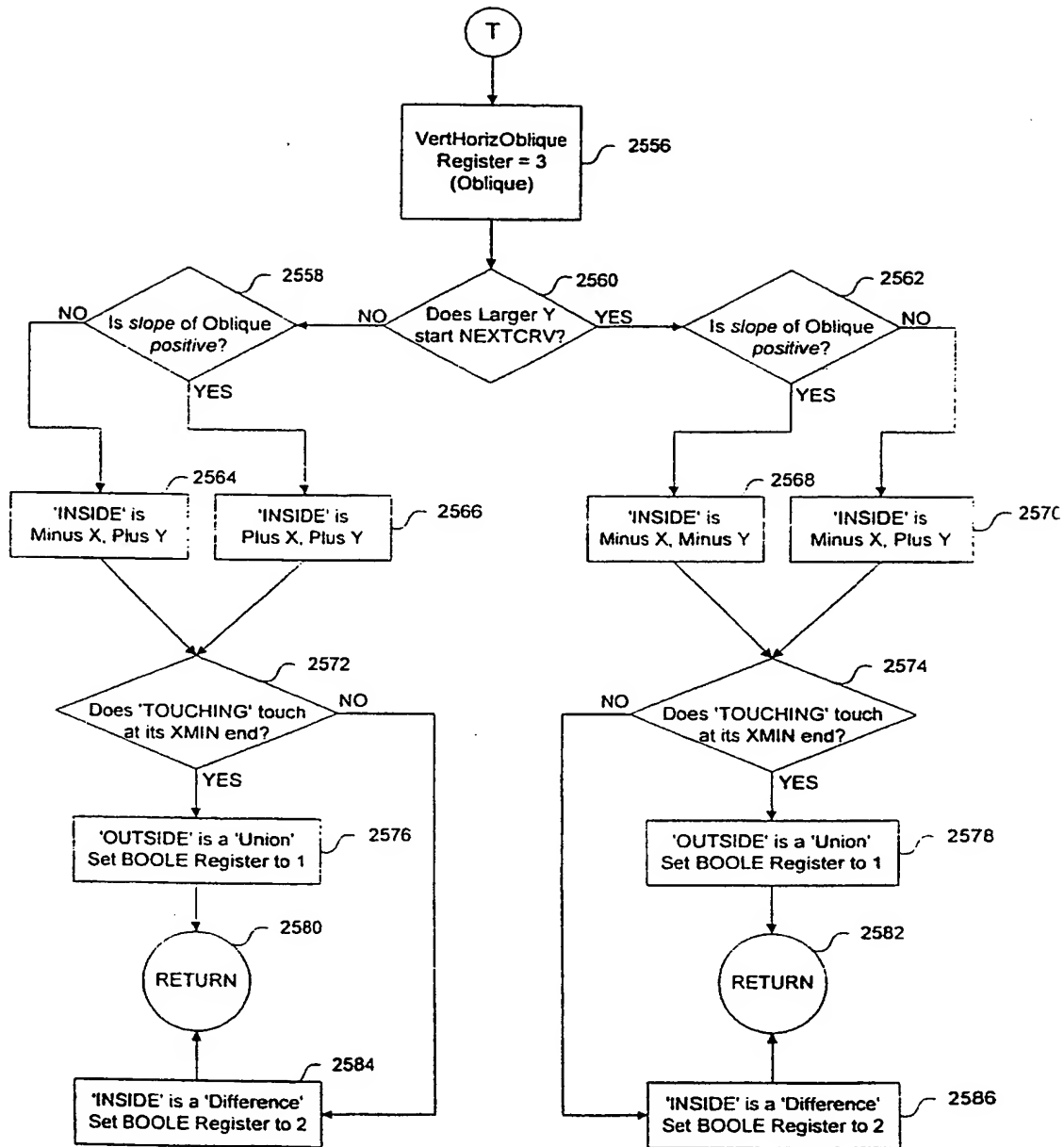


Fig. 15P

62/64

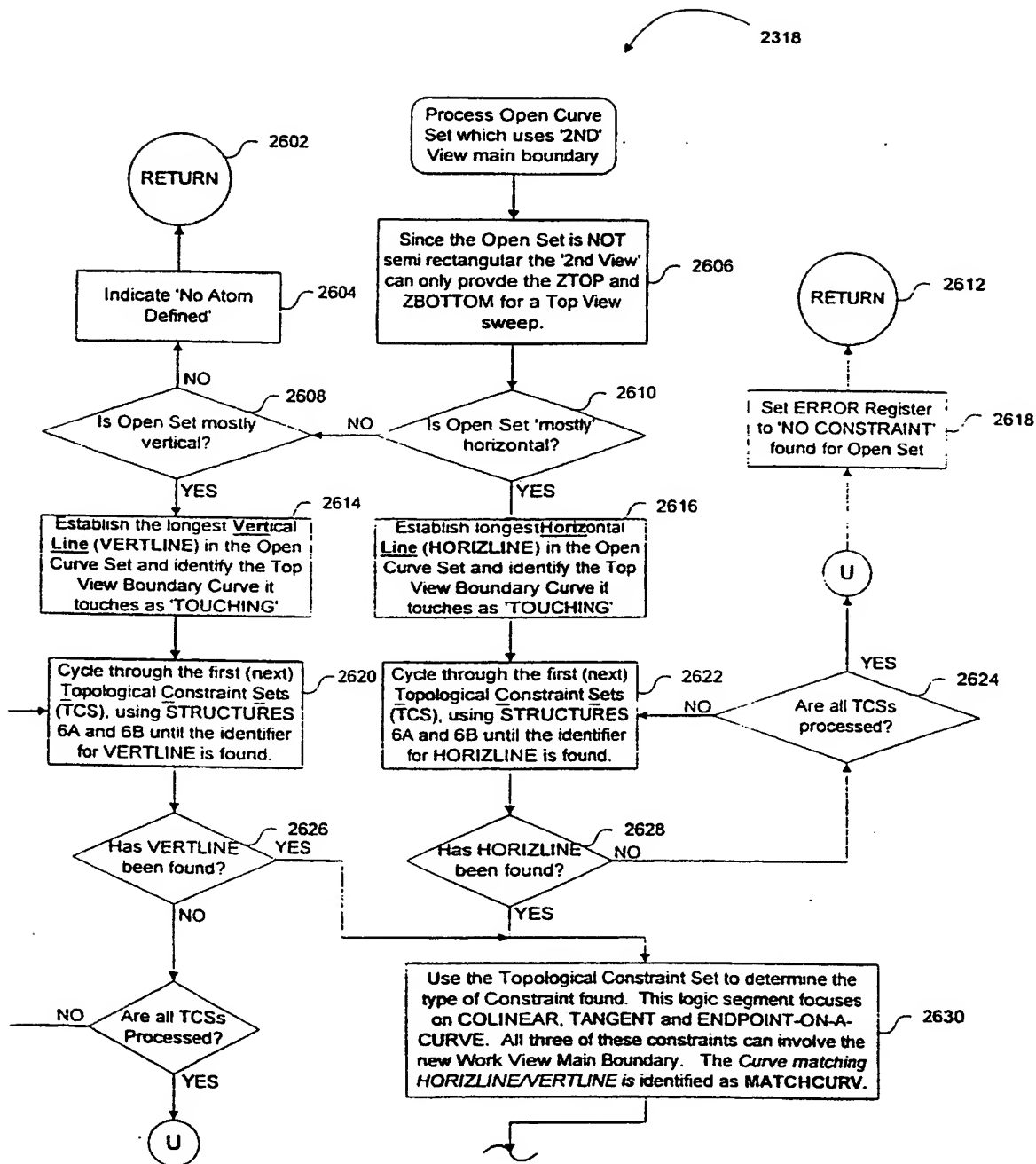


Fig. 15Q

63/64

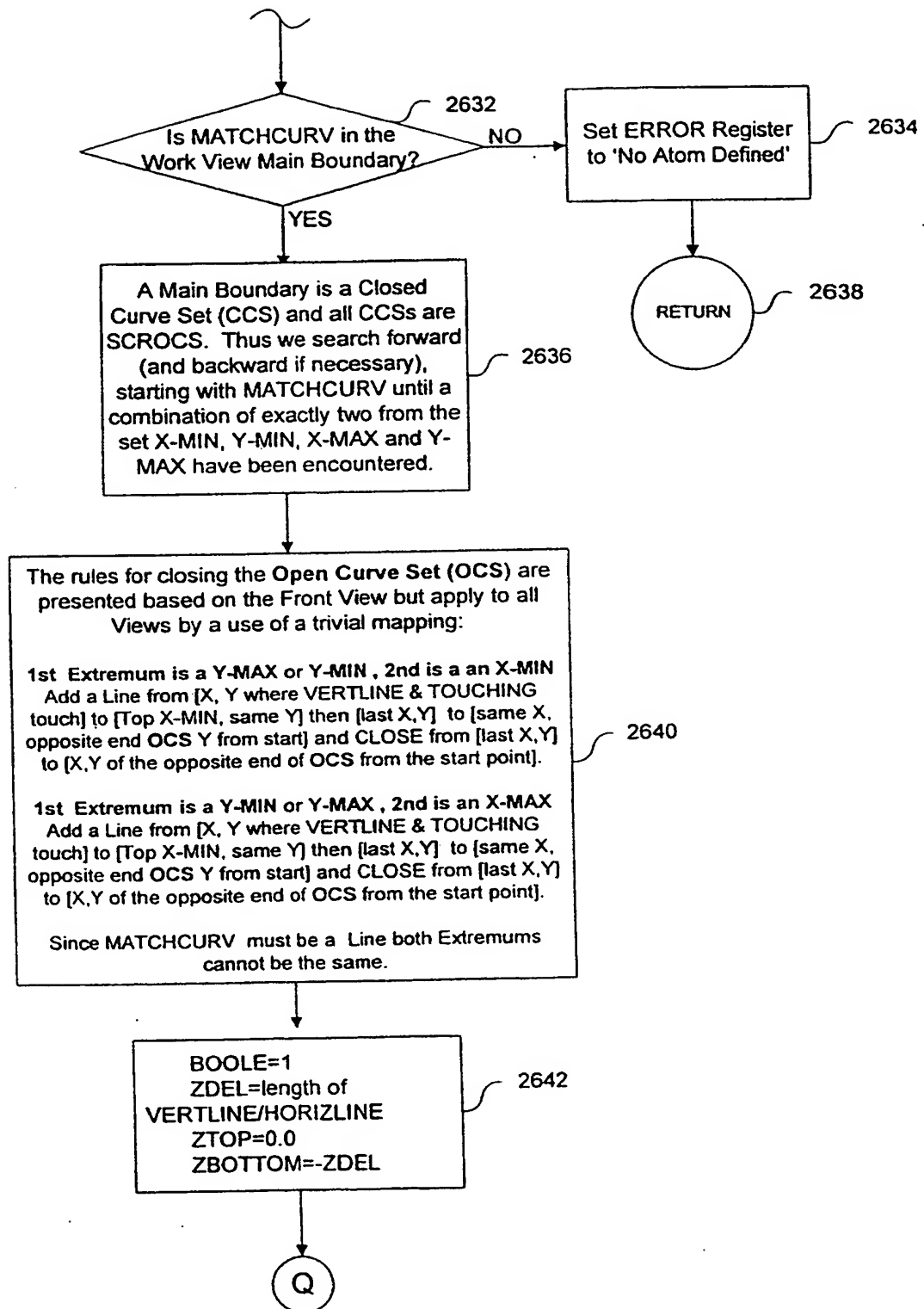


Fig. 15 R

64/64

INTERNATIONAL SEARCH REPORT

Internatio.

PCT/US98/19189

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : G06T 17/00, 17/40; G06K 9/36

US CL : 345/420, 427; 382/288

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 345/420, 427; 382/288

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

Please See Extra Sheet.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ALDEFELD, B. On Automatic Recognition of 3-D Structures from 2-D Representations. Computer Aided Design. March 1983, Vol. 15, No. 2, pp. 59-64, especially page 61, col. 1, and page 62, col. 2.	2, 7-8, 10
X	SUNDARAMURTHY, V. et al. Reconstruction of 3D CAD Models from 2D Orthographic Views. Proc. 4th International Conference on CAD, CAM, Robotics and Automous Factories, October 1994, pages 71-80, especially page 72, Figs. 2-3, and page 74.	1,3, 6, 16-17 ----- 18,19
Y		
X	ANONYMOUS. MakeIT3D 1.0: Reference Manual and Tutorial. published by EMT Software, Inc., December 1995, pages 1-91, especially pages 9, 14-15, and 23.	2-5 ----- 19
Y		

☒ Further documents are listed in the continuation of Box C.
 ☐ See patent family annex.

* Special categories of cited documents:	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
A document defining the general state of the art which is not considered to be of particular relevance	*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
E earlier document published on or after the international filing date	*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*A* document member of the same patent family
O document referring to an oral disclosure, use, exhibition or other means	
P document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

01 DECEMBER 1998

Date of mailing of the international search report

05 FEB 1999

 Name and mailing address of the ISA/US
 Commissioner of Patents and Trademarks
 Box PCT
 Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

JEFFREY ALLEN ROSSI

Telephone No. (703) 305-3900

Form PCT/ISA/210 (second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

Internatic.
PCT/US98/19189

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	- US 5,663,883 A (THOMAS et al.) 02 September 1997	18
A	- US 4,858,150 A (AIZAWA et al.) 15 August 1989.	1-19
A,P	- US 5,671,343 A (KONDO et al.) 23 September 1997	1-19
A,P	- US 5,745,117 A (MASUDA et al.) 28 April, 1998.	1-19
A,P	- US 5,821,941 A (MILLSTEIN) 13 October 1998.	1-19
A	- GANESAN, R. et al. Flexicad: an Architecture for Integrated Product Modeling and Manufacture by Features. Proc. 4th Intl. Conference on Computer Integrated Modeling and Manufacturing Technology, October 1994, pages 60-65.	1-19
A	- BIN, HO. Inputting Constructive Solid Geometry Representations Directly from 2-D Orthographic Views. Computer Aided Design, April, 1986, Vol. 18, No. 3, pages 147-155.	1-19
A	- SAKURAI, Hiroshi et al. Solid Model Input through Orthographic Views. Association for Computing Machinery Computer Graphics. July 1983, Vol. 17, No. 3, pages 243-252.	1-19

INTERNATIONAL SEARCH REPORT

Internatic

PCT/US98/19189

B. FIELDS SEARCHED

Electronic data bases consulted (Name of data base and where practicable terms used):

APS file= 'USPAT'; DIALOG (TECHNICAL); I.E.E.E. JOURNALS+; PATENT CITATION INDEX; INTERNET;
Software Patent Institute

search terms: views, cad, orthogonal, transform, 2-D, 3-D, convert, open, closed, disjoint, curves

